

◆ 1. Understand the Basics of Dynatrace

Start with the foundational concepts:

- **What is Dynatrace?**
 - **OneAgent:** How it works and what it monitors.
 - **Smartscape topology:** Visualizing dependencies between services, processes, and hosts.
 - **Davis AI:** Dynatrace's AI engine for root cause analysis.
-

◆ 2. Monitoring Types to Focus On

Infrastructure Monitoring

- Hosts, CPU, memory, disk, and network usage.
- Monitoring **Linux/Windows servers**.
- **Custom metrics** and **thresholds**.

Database Monitoring

- Supported databases (e.g., Oracle, MySQL, PostgreSQL).
- Query performance, slow queries, connection pools.
- Database service flow and dependencies.

Synthetic Monitoring

- **HTTP Monitors:** For API and endpoint availability.
- **Browser ClickPath Monitors:** Simulate user interactions in a browser.
- Alerting and SLA tracking.

Application Performance Monitoring (APM)

- Service-level monitoring (Java, .NET, Node.js, etc.).
 - Distributed tracing.
 - Error and exception tracking.
-

◆ 3. Key Features to Learn

- **Dashboards:** Creating custom dashboards for different teams.
 - **Problem Detection:** How Dynatrace detects and reports issues.
 - **Alerting & Notifications:** Integration with email, Slack, Teams, etc.
 - **Tagging and Management Zones:** For organizing and filtering data.
-

◆ 4. Hands-On Practice

- Set up a **Dynatrace trial account** (if not already provided).
 - Deploy **OneAgent** on a test server.
 - Create synthetic monitors.
 - Explore Smartscape and service flow maps.
 - Simulate issues and observe how Dynatrace detects them.
-

◆ 5. Learning Resources

- Dynatrace University: Free courses and certifications.
 - Dynatrace Documentation: Official and detailed.
 - YouTube tutorials and community forums.
-

◆ 6. Optional but Useful

- **APIs:** For automation and custom integrations.
- **Extensions:** For monitoring technologies not supported out-of-the-box.
- **Log Monitoring:** If your project includes log analysis.

What is Dynatrace?

Dynatrace is an **all-in-one observability and application performance monitoring (APM) platform** designed to help organizations ensure the health, performance, and reliability of their applications and infrastructure.

What Dynatrace Does

Dynatrace provides **real-time insights** into:

- **Applications** (web, mobile, microservices)
- **Infrastructure** (servers, hosts, containers, cloud platforms)
- **User experience** (synthetic and real user monitoring)
- **Logs and events**
- **Network and process-level metrics**

Key Features

1. **OneAgent**: A single agent that automatically discovers and monitors everything from the OS level to applications and services.
2. **Smartscape**: A real-time topology map showing how all components are connected.
3. **Davis AI**: An AI engine that automatically detects anomalies, pinpoints root causes, and reduces alert noise.
4. **Synthetic Monitoring**: Simulates user interactions to test availability and performance.
5. **Real User Monitoring (RUM)**: Tracks actual user behavior and experience.
6. **Infrastructure Monitoring**: Monitors CPU, memory, disk, and network usage across hosts and containers.
7. **Log Monitoring**: Collects and analyzes logs for deeper troubleshooting.

Use Cases

- Detect and resolve performance issues quickly.
- Monitor cloud-native environments (Kubernetes, AWS, Azure, GCP).
- Ensure SLAs and uptime for critical services.
- Gain visibility into user journeys and optimize UX.
- Automate root cause analysis and reduce MTTR (Mean Time to Resolution).

What is OneAgent in Dynatrace?

OneAgent is the **core component** of Dynatrace. It's a lightweight software agent that you install on your **hosts (servers, VMs, containers, etc.)**. Once installed, it **automatically detects and monitors** everything running on that host—**applications, services, processes, containers, databases, and infrastructure metrics**—without requiring manual configuration.

What OneAgent Does:

- **Auto-discovers** all running processes and services.
 - Collects **metrics** (CPU, memory, disk, network).
 - Captures **traces** of transactions across services.
 - Monitors **logs** and **events**.
 - Supports **cloud-native** environments (Kubernetes, AWS, Azure, etc.).
-

Example Scenario

Let's say you have a **Linux server** running:

- A **Java-based web application**
- A **MySQL database**
- An **NGINX web server**

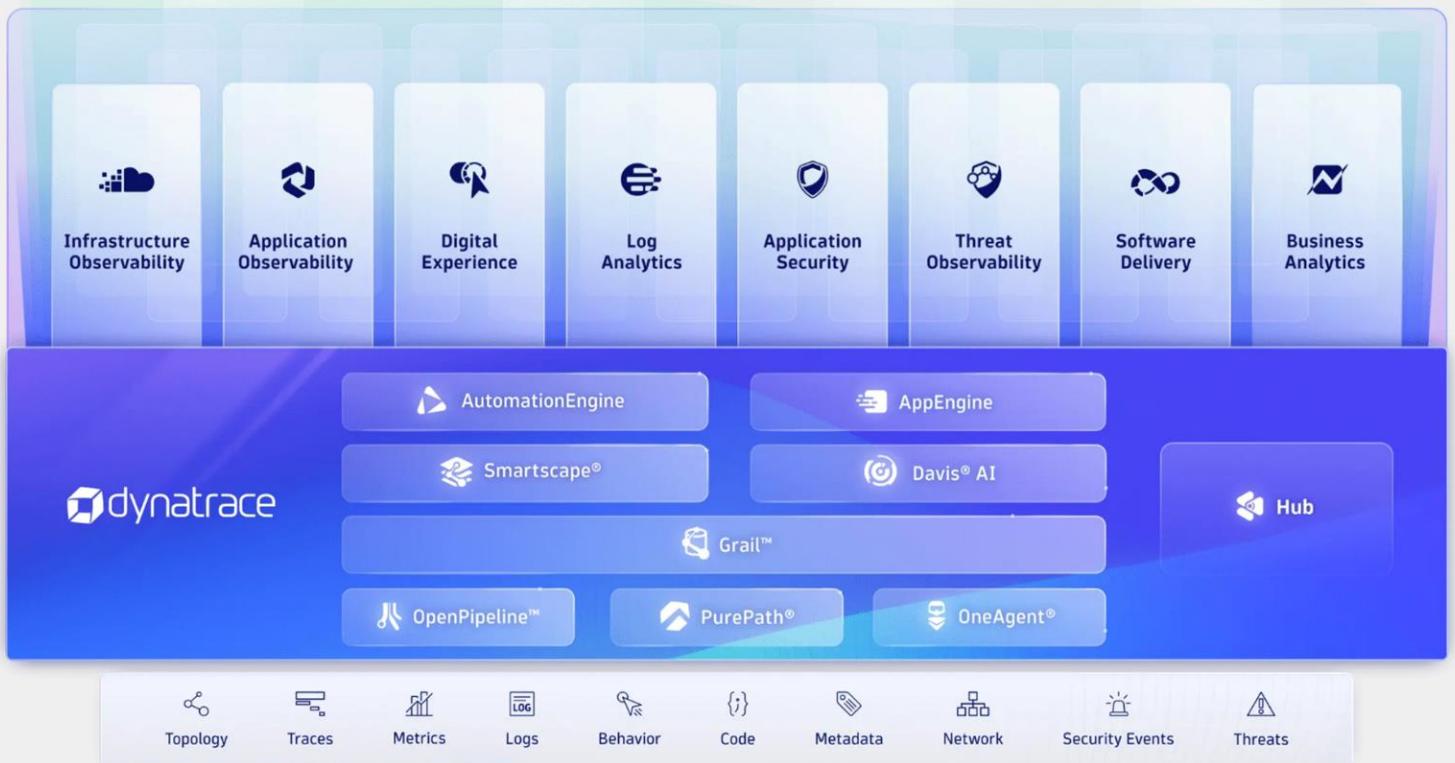
Once you install **OneAgent** on that server:

1. It detects the **host OS** and starts collecting CPU, memory, disk, and network metrics.
2. It identifies the **Java process**, instruments it, and starts tracing requests.
3. It detects the **MySQL database**, monitors query performance and connections.
4. It sees the **NGINX server**, tracks incoming HTTP requests and response times.
5. It maps how these components interact using **Smartscape**.

All of this happens **automatically**, without needing to configure each component manually.



- Answers and intelligent automation—Dynatrace uses causal AI to automate DevSecOps at scale and deliver the precise answers teams need to innovate and thrive in the modern cloud.



Synthetic Monitoring

Create a synthetic monitoring

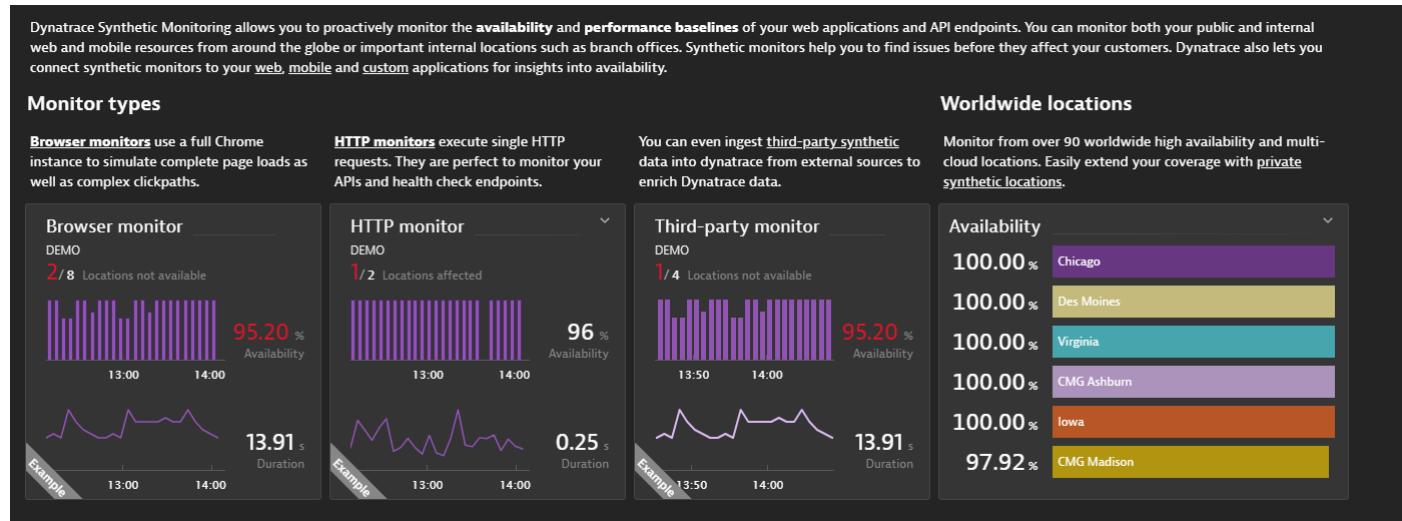
1. There are two types of monitoring.

- a. Browser Monitoring
- b. HTTP Monitoring.

Browser Monitoring

1. Install Dynatrace Synthetic Recorder
2. Manually add clickpath events.

Tags are used for referencing the objects.



In **Dynatrace**, the difference between **Full-Stack Monitoring** and other types of monitoring (like **Infrastructure Monitoring** or **Application Monitoring**) lies in the **depth and scope** of visibility into your systems. Here's a breakdown:

🔍 Full-Stack Monitoring (FSM)

Definition: Provides **end-to-end visibility** across the entire technology stack — from frontend user interactions to backend infrastructure.

Includes:

- **Real User Monitoring (RUM):** Tracks actual user sessions and behavior.
- **Application Performance Monitoring (APM):** Deep code-level insights into services, APIs, and transactions.
- **Infrastructure Monitoring:** Metrics from hosts, containers, VMs, and cloud services.
- **Log Monitoring:** Centralized log collection and analysis.
- **Network Monitoring:** Tracks traffic between services and dependencies.
- **Synthetic Monitoring:** Simulated user interactions to test availability and performance.

Use Case: Ideal for DevOps, SREs, and developers who need **complete observability** to troubleshoot issues across the entire stack.



Other Monitoring Types in Dynatrace

1. Infrastructure Monitoring Only

- Focuses on **hosts, VMs, containers, and cloud infrastructure.**
- No visibility into application code or user experience.
- Lower licensing cost.
- Suitable for teams only interested in system health and resource usage.

2. Application Monitoring Only

- Monitors **application performance** and **code-level traces.**
- May not include full infrastructure or user session data.
- Useful for developers focusing on app behavior.

3. Digital Experience Monitoring (DEM)

- Focuses on **user experience** via RUM and Synthetic Monitoring.
 - Doesn't include backend or infrastructure insights.
-

vs Summary Table

Feature	Full-Stack Monitoring	Infrastructure Monitoring	Application Monitoring	DEM
User Session Tracking (RUM)	✓	✗	✗	✓
Code-Level Tracing (APM)	✓	✗	✓	✗
Host/VM/Container Metrics	✓	✓	✗	✗
Synthetic Monitoring	✓	✗	✗	✓
Log Monitoring	✓	✓ (limited)	✓ (limited)	✗

Great question! **Grafana** plays a complementary role in the observability ecosystem, and it can integrate with **Dynatrace** to enhance visualization and dashboarding capabilities.

Here's how it fits in:

⌚ Grafana's Role in Monitoring

Grafana is an **open-source analytics and visualization platform**. While Dynatrace provides **deep observability and AI-powered insights**, Grafana is often used for:

✓ Custom Dashboards

- Grafana excels at creating **highly customizable dashboards**.
- You can pull data from Dynatrace (via API or plugins) and combine it with other sources like **Prometheus, InfluxDB, Elasticsearch**, etc.

🔗 Multi-Source Correlation

- Grafana allows you to **correlate metrics from multiple tools** in one place.
- For example, you can visualize Dynatrace APM data alongside infrastructure metrics from Prometheus.

Advanced Visualization

- Grafana supports a wide range of **visualization panels** (graphs, heatmaps, gauges, etc.).
- Useful for **executive dashboards, NOC views, or custom SLA reporting**.

Access Control & Sharing

- Grafana provides **fine-grained access control**, making it easier to share dashboards with different teams or stakeholders.

Grafana + Dynatrace Integration

You can integrate Dynatrace with Grafana using:

- **Dynatrace plugin for Grafana** (official or community-supported)
- **Dynatrace API** to pull metrics and logs into Grafana panels

When to Use Grafana with Dynatrace

Use Case

Use Dynatrace Use Grafana

Deep root cause analysis		
AI-powered anomaly detection		
Custom dashboards across tools		
Executive or NOC dashboards		
Unified view from multiple data sources		

OneAgent vs Davis AI

Great question! In Dynatrace, OneAgent and Davis AI are two core components, but they serve very different purposes. Here's a clear comparison:

Davis AI (Dynatrace AI Engine)

Role: Analyzes and correlates data collected by OneAgent and other sources to detect problems, anomalies, and root causes.

Key Features:

- AI-powered root cause analysis
- Automatic anomaly detection
- Smart alerting (avoids alert storms)
- Dependency mapping (understands service relationships)
- Predictive analytics (in some tiers)

Think of it as: The brain of Dynatrace that makes sense of all the data.

OneAgent

Role: Collects data from your environment — applications, services, processes, hosts, containers, and more.

Key Features:

- Installed on hosts (VMs, bare metal, containers)
- Captures metrics, traces, logs, and events
- Enables auto-instrumentation of applications
- Supports Full-Stack Monitoring

Think of it as: The sensors or eyes and ears of Dynatrace.

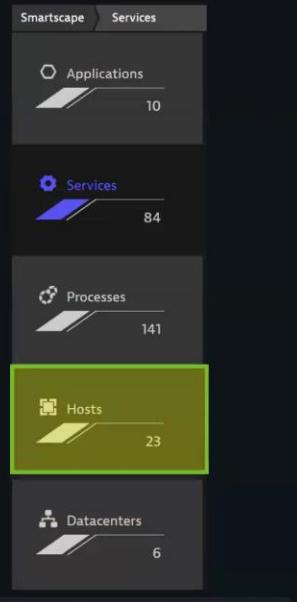
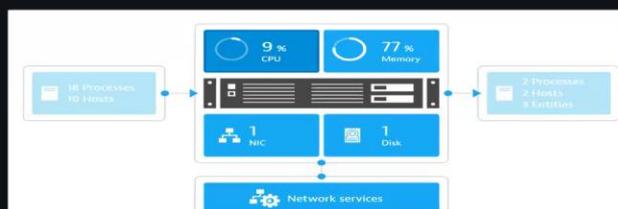
vs **Summary Table**

Feature	OneAgent	Davis AI
Purpose	Data collection	Data analysis & correlation
Installed on	Hosts, containers, VMs	Not installed (cloud-native AI)
Collects	Metrics, logs, traces, events	Uses collected data
Detects anomalies	✗	✓
Performs root cause analysis	✗	✓
Sends alerts	✗	✓
Required for monitoring	✓	✓ (to analyze OneAgent data)

Would you like a visual diagram showing how OneAgent and Davis AI interact in the Dynatrace architecture?

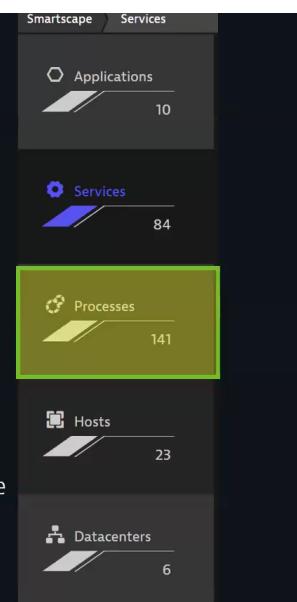
The Host Layer

- A physical or virtualized operating system
- The source of compute, memory, and storage resources
- Host related metrics reside here including:
 - CPU
 - Memory (RAM)
 - Network interface Cards (NICs)
 - Disk Storage and Utilization



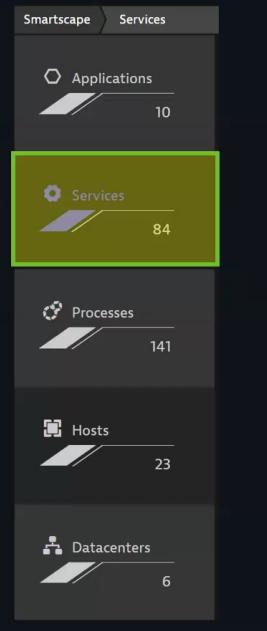
The Process Layer

- A currently executing computer program on a host
 - Think Java, .NET, Jetty Processes
- A means for code to request computing resources
- Metrics include:
 - Process CPU
 - JVM Heap
 - PID
 - Threads
- Processes are both tangibly and logically grouped in Dynatrace.
 - We have 3 Java application JVMs across 3 hosts = Tangible=Process group Instance
 - Those 3 Java JVMs make up my Prod Application server= Logical= Process Groups



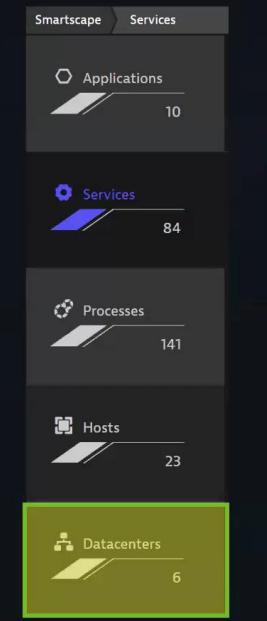
The Service Layer

- A set of code or classes that performs some type of business logic
- Services are logical groupings that run on a Process or Processes
- Transactions occur at this layer
 - Web requests
 - Web services
 - Database requests
- Metrics include:
 - Failure rate
 - Response time
 - Throughput
 - Exceptions



The Datacenter Layer

- A physical location on Earth where hosts reside.
- An organization can have one or multiple datacenters in use
- Related terms: vCenter, zone, Virtual, AWS, Geo Region



The Application Layer

- The point that End users interact with Services to perform business logic.
- Think User analytics
- Metrics in this layer include:
 - Browser
 - User action time
 - W3c timings
 - Conversion goals
 - Location distribution
- Applications are logical Groupings based on URL or Domain



What are Services In Practice?

- Services are where transactions occur for your application
 - Example: Your travel website has functions/requests (add to cart, sign in, search etc.)
 - The Customer front end service contains these functions or requests **within the service**
- Services often call other services to get the task or request completed.
 - Search requests usually call a database service
- Services usually fall under common types:
 - Web requests
 - Web services
 - Database Calls
 - Message Queues
 - A custom entry point

Intro to Dynatrace

Key-Terms

1. Applications
2. Services
3. Processes
4. Hosts
5. Data centers

So understand the overview

Great! Let's explore the **relationship and comparison** between these five components in the context of **IT infrastructure**. Here's how they typically interact and differ:

1. Applications

- **What they are:** Software programs designed to perform specific tasks for users or other applications.
 - **Examples:** Web browsers, CRM systems, ERP software, mobile apps.
 - **Dependency:** Run on top of services and processes, often distributed across multiple hosts or data centers.
-

2. Services

- **What they are:** Background components or functions that support applications or system operations.
 - **Examples:** Web servers (e.g., Apache), database services (e.g., MySQL), authentication services.
 - **Relationship:** Applications rely on services to function (e.g., a web app needs a database service).
-

3. Processes

- **What they are:** Instances of running programs (including services and applications).
 - **Examples:** A running instance of nginx, java, or python.
 - **Relationship:** Services and applications are made up of one or more processes. Processes are the execution units on a host.
-

4. Hosts

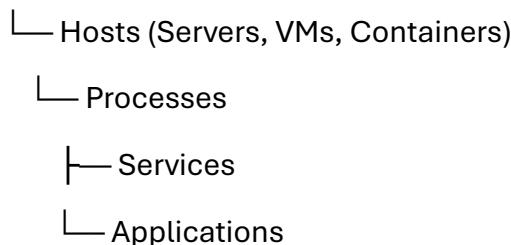
- **What they are:** Physical or virtual machines that run processes.
 - **Examples:** Servers, desktops, virtual machines, containers.
 - **Relationship:** Hosts provide the environment for processes, services, and applications to run.
-

5. Data Centers

- **What they are:** Physical facilities that house hosts (servers), networking equipment, and storage.
 - **Examples:** AWS regions, Google Cloud zones, on-premise server rooms.
 - **Relationship:** Data centers contain the infrastructure (hosts) that run all the above components.
-

How They Interact (Hierarchy View)

Data Centers



Dashboards and Data Explore

Visualization

Host, CPU memory in graphs format

We need write access, Generic one we can explore.

The **Data Explorer** in Dynatrace is a powerful tool designed to help you **query, analyze, and visualize metrics** from your environment. Here's a breakdown of what it offers and how it fits into your infrastructure monitoring:

Data Explorer is a **visual query builder** that allows you to:

- Explore **built-in and custom metrics** (from sources like StatsD, Prometheus, Telegraf, etc.).
- Create **multidimensional queries** using filters, aggregations, and splits.
- Visualize data using charts like **line graphs, bar charts, heatmaps**, and more.
- Pin visualizations to **classic dashboards** for ongoing monitoring and sharing

Each query in Data Explorer can include:

- **Metric name** (e.g., builtin:host.cpu.usage)
- **Aggregation** (e.g., average, sum, percentile)
- **Split by** (e.g., host, service, process)
- **Filters** (e.g., OS type = Linux)
- **Timeframe** (e.g., last 30 minutes, custom range)

Use Cases

- **Compare performance** across hosts, services, or applications.
- **Track trends** over time (e.g., CPU usage, memory consumption).
- **Identify anomalies** or spikes in metrics.
- **Create dashboards** for different teams or use cases.

Advanced Features

- **:timeshift transformation**: Compare metrics across different timeframes (e.g., this week vs. last week).
- **Rate and unit transformations**: Convert raw counts into rates (e.g., requests per minute) and set appropriate units

Hosts and Processes

The Host Layer

- A physical or virtualized operating system
- The source of compute, memory, and storage resources
- Host related metrics reside here including
 - CPU
 - Memory (RAM)
 - Network interface Cards (NICs)
 - Disk storage and utilization

The Process Layer

- A currently executing computer program on a host
 - Java, .NET, Jetty Processes
- A means for code to request computing resources
- Metrics include:
 - Process CPU
 - Heap
 - Threads
- Processes are both tangibly and logically grouped
 - We have 3 JVMs across 3 hosts = tangible (process group **instances**)
 - Those 3 JVMs make up my production application server = logical (process group)

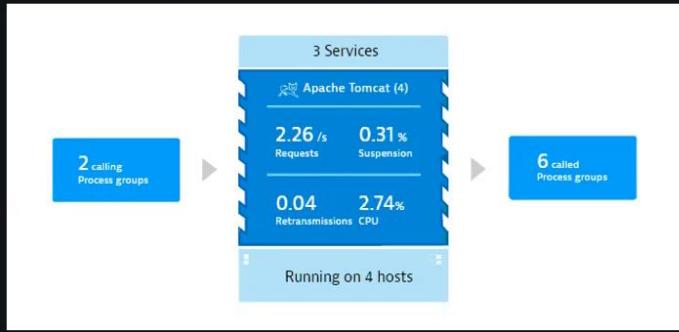


What are process group instances?

- A single tangible process that belongs to a designated process group
- Tangible - identified as running on one unique host, container, etc.
- Metrics include
 - Process CPU
 - Heap
 - Threads

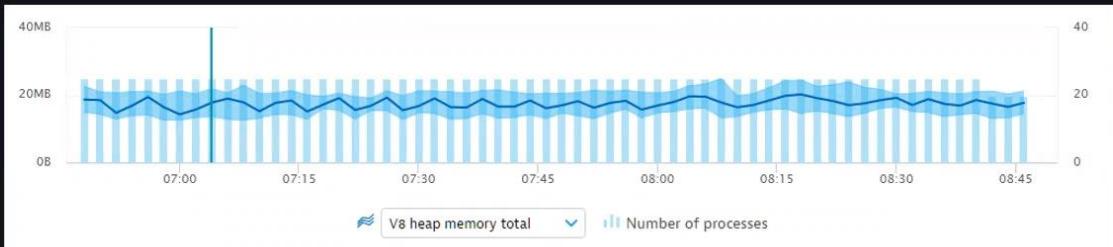
What are process groups?

- Organizational function
- Must run the same software technology and perform the same functions
 - Think horizontal scaling



What are process groups?

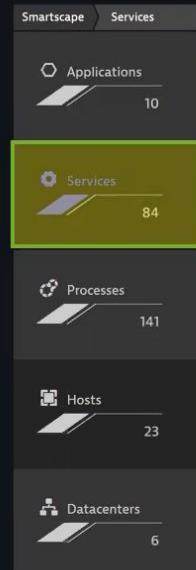
- Logical groupings
 - Dynatrace groups automatically, but may need to be configured
- Most important configuration
- Process groups are made up of process group instances.
 - Metrics together or separately



Service

The Service Layer

- A set of code or classes that performs some type of business logic
- Services are logical groupings that run on a Process or Processes
- Transactions occur at this layer
 - Web Requests
 - Web Services
 - Database requests
- Metrics include:
 - Response Time
 - Failure Rate
 - Throughput
 - Exceptions



What are Services In Practice?

- Services are where transactions occur for your application
 - Example: Your travel website has functions/requests (add to cart, sign in, search etc.)
 - The Customer front end service contains these functions or requests **within the service**
- Services often call other services to get the task or request completed.
 - Search requests usually call a database service
- Services usually fall under common types:
 - Web requests
 - Web services
 - Database Calls
 - Message Queues
 - A custom entry point

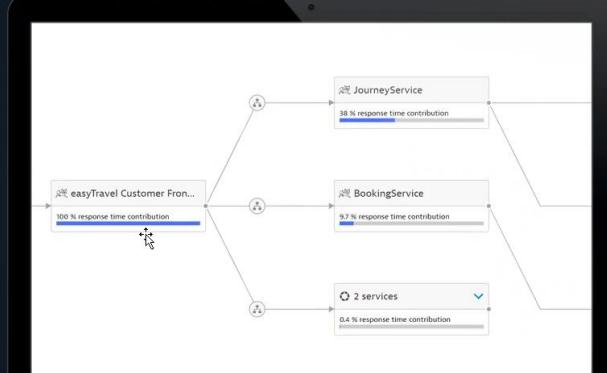
What do we get with Service monitoring?

- Monitoring begins automatically as soon as Dynatrace OneAgent starts operation and you restart all your application's server processes
- Dynatrace monitoring of services extends all the way down to the monitoring of discrete methods (code) and code level information is collected for each request, known as Purepaths
- Every Service (and transaction within) has a dynamic baseline calculated that considers Response time and Failure rate which is "normal" for your service in question.
- All services views, baselines, and metrics are derived from one or many PurePaths.
- Services can be renamed and merged to fit your use cases.

The screenshot shows a mobile device displaying the Dynatrace application. The top navigation bar is blue with the Dynatrace logo. The main screen is titled 'Top requests' and features a table with several rows of data. The columns include 'Name', 'Icefaces', 'Client IP Ad...', and 'Content-Len...'. The rows list various URLs: '/special-offers.jsp', '/CalculateRecommendations', '/orange.jsf', and '/legal-orange-mobile.jsp'. Each row also includes a 'Time' column with values like 'easyTravelD...' and 'easyTravel D...'. The bottom right corner of the screen shows the number '5'.

Services – Cont.

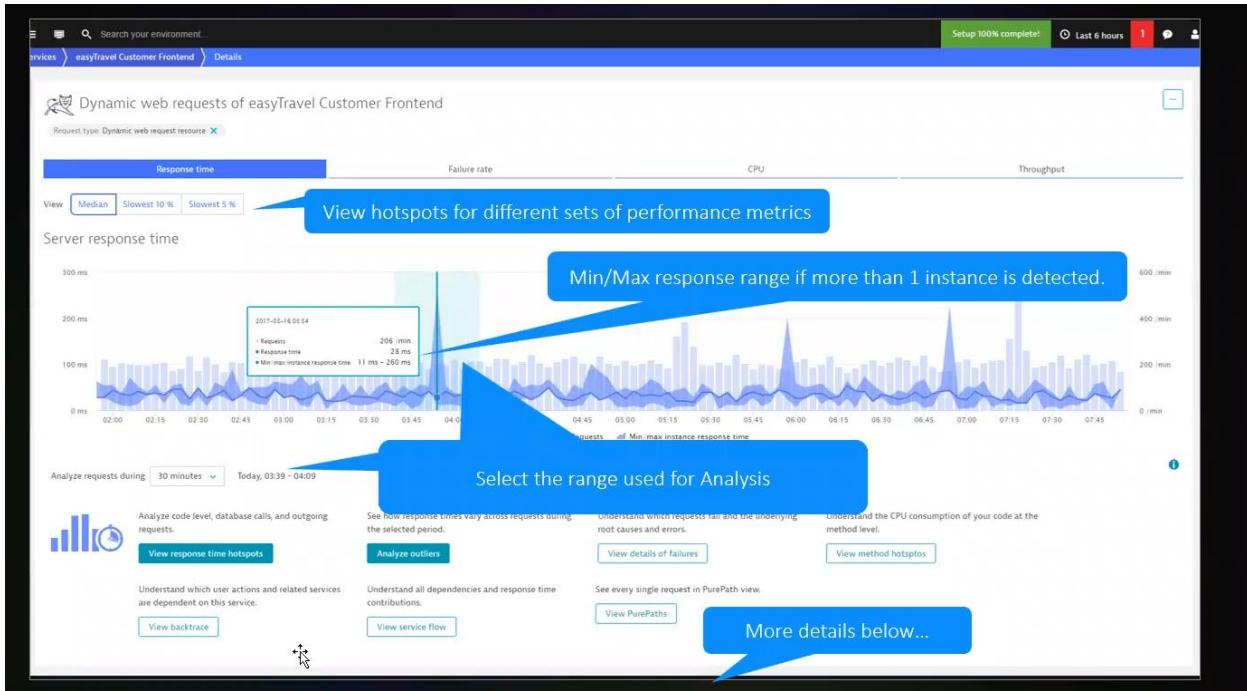
- Services are usually automatically detected, they include:
 - Web services
 - Web requests,
 - Database requests
 - Message queues
 - Custom services
- Services may call one or many other services to complete their transactions



Confidential 6

The screenshot displays a comprehensive monitoring interface for the easyTravel Customer Frontend:

- Infographic and Service overview:** Shows key metrics: Response time (5.42k/min), CPU (4.26 ms), Failure rate (0%), Throughput (673/min), and Network clients (1 Apache to.).
- Problem history:** Lists 3 problems in the last 72 hours.
- Hotspot Analysis:** Identifies current hotspots: High consumption of memory (23%) and Slow response time (242 ms).
- Drilldown into Dependencies:** Allows users to understand dependencies and analyze backtraces.
- Event logger:** Shows no events for today.
- Dynamic web requests:** Provides detailed metrics for response time, failure rate, CPU, and throughput.
- Resource requests:** Provides metrics for response time, failure rate, CPU, and throughput.



What are we covering during this call?

- Application
 - How are metrics collected
 - How we define an Application
 - Live Session
- Synthetics
 - Types of Synthetic Monitors
 - Typical Use Cases
 - Live Session
- Where to get help



Applications Overview

- The point that end users interact with Services to perform business logic (services)
- User Analytics
- Metrics in this layer include:
 - Browser
 - User action time
 - W3C timings
 - Key performance metrics
 - Conversion goals
 - Location distribution
- Logical grouping based on URL or domain

Dynatrace OneAgent identifies HTML content responses and automatically injects a small JavaScript tag into the head section of each page. You can alternatively inject the JavaScript tag into HTML code manually without installing Dynatrace OneAgent. We provide the JavaScript tag for you during the [set up](#) process. You can paste the tag into the pages you intend to monitor.

You can, for example, tell Dynatrace to inject the tag before or after a defined string in an HTML response. In the case of [Java](#) you can specify injection within the innermost or outermost `HttpServletResponse`, within a custom servlet, or within a filter.

The dashboard displays the following data:

Category	Value
Applications	10
Services	84
Processes	141
Hosts	23
Datacenters	6

How are Application metrics collected?

- In order to collect these user metrics, we must inject a small piece of JavaScript code into your web application in question
 - Usually handled automatically by the OneAgent if installed on the web server in question
 - The JavaScript snippet can be inserted manually if the OneAgent is not installed in question (aka Agentless)
- Once injected, Dynatrace will collect user metrics, browser information, and geolocation
- Some additional configuration may be needed. Reach out via Live chat for any questions

Application to Services Relationship

- End users interact with your organization/application via web browser, mobile app etc in a designated user session
- User sessions are made up of user actions (click on page, submit order, search etc)
- Each user action calls on services in order to complete the task
- Example: Logging into dynatrace.com, search "dynatrace one premium"
- User action= search "dynatrace one premium"
- Service called= Dynatrace.com database for "dynatrace one premium"
- Users are tracked in user sessions. User sessions are made up of user actions. User actions call services to perform business logic

Synthetics Overview

- What are Synthetics?
 - Simulation of clickpaths or HTTP Monitors for applications
 - Allows the testing of an application using many different locations across the world.
- 2 different types of Synthetic tests:
 - HTTP Monitors
 - Browser Clickpaths/Monitors
 - Public Locations
 - Private Locations

HTTP Monitors

- HTTP Monitors are the most basic of Synthetic tests
- An HTTP monitor uses a simple HTTP request to monitor the availability of a given URL
 - Example, validating an API endpoint is up and available across locations
- HTTP monitors are highly customizable in terms of what is sent in the said http request
 - Headers
 - Request body
 - Redirects
 - Response code handling
- Results are http response code oriented. No Waterfall charts are available.

Browser Clickpath Monitor

- Simulates a configured transaction
 - Can be one or many pages
- Response time and availability are recorded for every step as well as for the overall transaction
 - Full waterfall charts are available
- Runs on real browsers across each configured location(s)
- Requires a browser extension to record
 - Each test execution runs as a new user without any saved cookies or any saved browser cache
 - Records a screenshot for each failing test run
- Can be used to monitor internal applications
 - Requires synthetic-enabled ActiveGate

Recording a Clickpath Notes

- Once you start recording you just need to perform every action required for the test in the new browser window that is opened.
- All data is captured from the new windows and saved in the Dynatrace UI while recording.
 - Pages that are navigated to
 - CSS selectors and DOM details of elements that are interacted with
 - Text of forms that are filled
- To end the recording simply close the window where the recorder is executing
- At this point you will arrive at the list of events of the clickpath. You will need to continue to save the test. Leaving this page without saving will cause you to lose your clickpath

Meet Davis

- Davis, an AI causation engine automatically detects performance anomalies in your applications, services, and infrastructure.
- Davis constantly monitors your environment based on the hundreds of metrics collected by the OneAgent.
- Davis biggest benefits include:
 - Automatic Baselineing of all Services' response times and failure rates
 - Root cause isolation and presenting related metric anomalies found in your environment

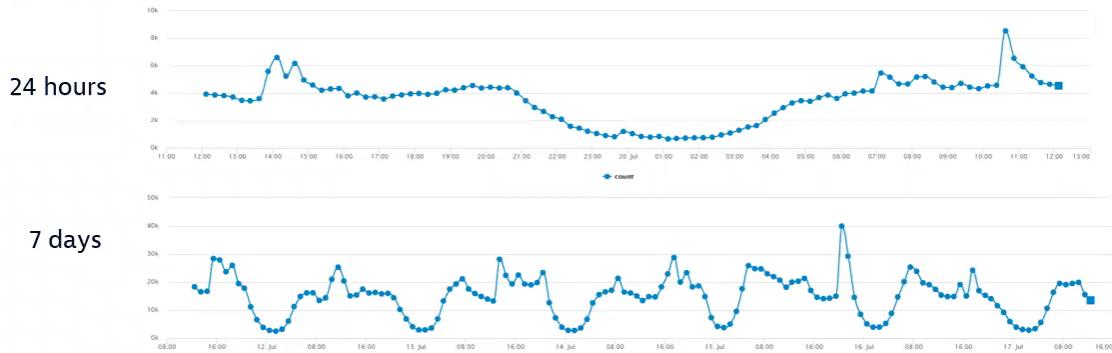


Events in Dynatrace

- In short, Dynatrace (Davis) notices related events in your environment, which become problems based on overall impact. Events are dictated through anomaly detection.
- Anomaly Detection exists based on two types of thresholds Davis compares against
 1. **Automated baselines:** Multidimensional baselining automatically detects individual reference values that adapt over time.
 2. **Built-in static thresholds:** Built-in static thresholds for all infrastructure events (for example, detecting high CPU, low disk space, or low memory)
- Dynatrace offers complete customization of both threshold types, but we recommend:
 1. Letting Davis monitor your environment and automatically determine baselines for your RUM & Service data
 - Special cases (Key Requests, SLA tracking, etc.) exist to give you complete control as needed
 2. Static Thresholds for Infrastructure can be customized to fit your environment and needs

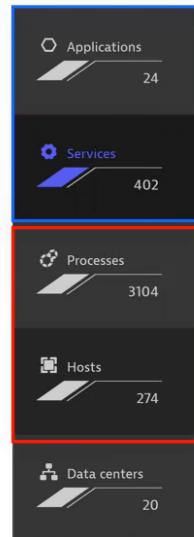
Anomaly Detection

- Dynatrace monitors the baseline performance and behavior of applications, services, infrastructure components etc.
- Dynatrace compares the current values with the values 24 hours previously and 1 week previously



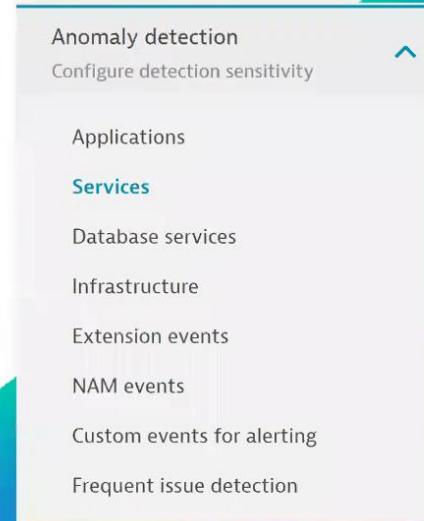
Baseline or Static Threshold for anomaly detection?

- It is important to understand which key metrics fall under which threshold by default.
- A simple general guide to follow on what metrics are baselined vs which are static is to refer to the [Smartscape](#).
- The Application and Service Layer Metrics are automatically baselined.
- Process, Host, and Datacenter metrics use Built-in Static Thresholds
- You have complete control to change these thresholds, but we recommend you let Davis decide your automated baselines.



Adjusting the sensitivity of Anomaly Detection

- Dynatrace has default thresholds for anomaly detection
- These setting can be changed under settings > Anomaly detection



Adjusting the sensitivity of Anomaly Detection (Applications Example)

- We can change the anomaly detection settings by going to Settings > Anomaly detection > Applications

Response time

Detect key performance metric degradations

Detection strategy for key performance metric degradations

Automatic

All user actions

Alert if the average response time of all user actions degrades beyond **both** the absolute and relative thresholds:

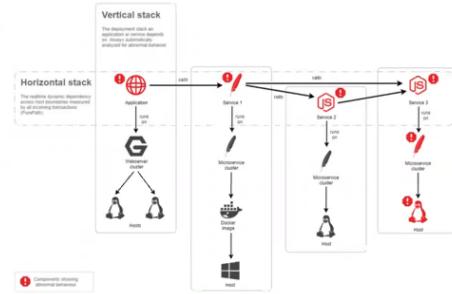
Absolute threshold: 100 ms

Relative threshold: 50 %

- Dynatrace distinguishes between an absolute threshold and a relative threshold for the median and the slowest 10 percent of each given metric

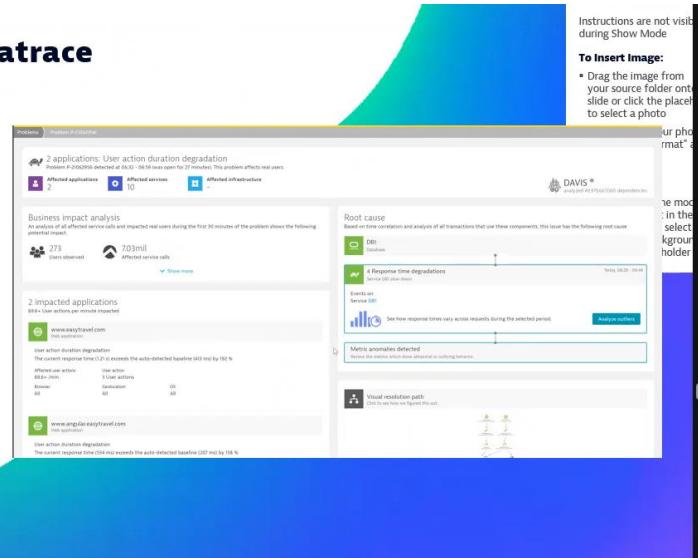
What happens when Davis opens a Problem?

- Davis notices an Event and finds all related components
- Davis follows transactions and collects evidence metrics
- Davis opens a problem, designates on a Severity
- Severities Include:
 - Availability
 - Error
 - Slowdown
 - Resource
 - Custom



How to interpret Problems in Dynatrace

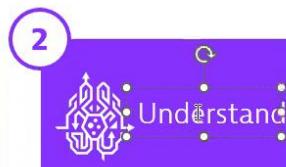
- When Davis opens a Problem, it will present all the needed information in one place.
- It will also trigger a problem notification if configured.
- Taking this information, understanding it and acting on it will give you the biggest value in using Dynatrace.
- Let's review 3 common problems in Dynatrace
 - Resource Contention
 - Response degradation
 - Failure rate increase



Keys to Dynatrace Problem Resolution Success



Once DAVIS notifies you of a problem, review the generated problem card to understand the Severity and how long said problem has been open.



Understand the impact in terms of the Smartscape. Review any metric anomalies presented in the problem card as well. Read up on the Root cause presented in the problem.



Let Davis guide you to the appropriate menus as it pertains to your particular problem for even more details.



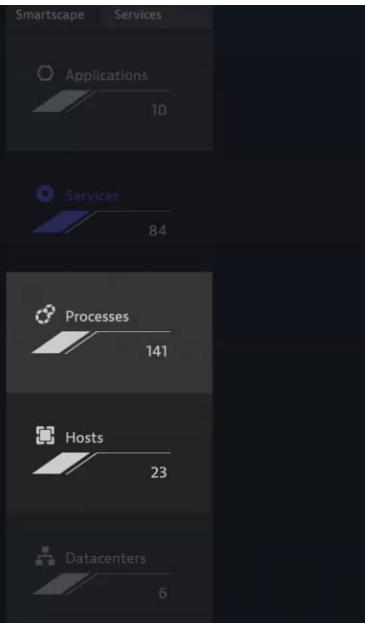
Copy links, grab screenshots and share with the appropriate teams to give them the needed information quickly

So, let's review!

- Davis, Our AI engine collects interprets hundreds of metrics and finds events based on both automatic and built-in thresholds
- Events get correlated into Problems, which Davis will then try to find the root cause of said problem based on all the metrics
- Keys to Dynatrace Problem Resolution Success
 - Review the Problem in Dynatrace
 - Understand the Impact Davis is seeing
 - Investigate further using our guided views
 - Share with your app teams as needed
- Problem discovery, review, and interpretation is the biggest value add of Dynatrace.

Concept review: Hosts and Process Groups.

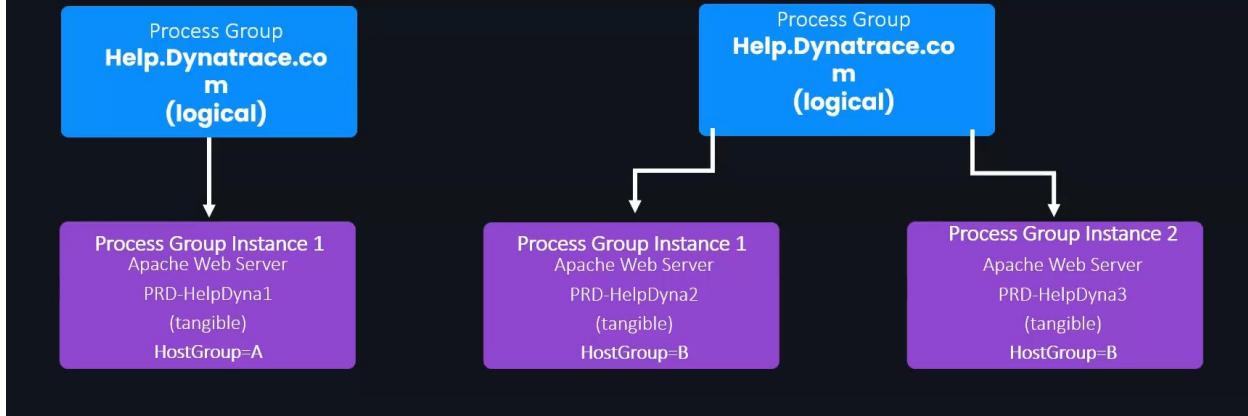
- In order to get the most out of this webinar, we need to have a solid understanding of Hosts and Process Groups
- Understanding and applying the SmartScape and its layers aides in understanding the difference and functionality between these groups.
- Always remember the following today:
 - All processes in Dynatrace belong to Process Groups.
 - All hosts belong to a host group (even if there is no Host Group defined)
 - Host Groups segment over Process Groups
 - Grouping is the most important configuration in Dynatrace



Host Groups vs Process Groups

- Host Groups (Host Layer)
 - A logical grouping and segmentation of hosts that serve the same function or grouping to fit your needs
 - A host can only apply to only one host group
 - Host groups can be by app, OS, function depending on your use case
 - Host Groups Influence Process Groups
- Process Groups (Process Layer)
 - A logical grouping based on some inherit or unique Process property
 - Environment Variable
 - Command line argument
 - Technology specific
 - All Process belong to a Process Group
 - All processes in a process group must all do the same function
 - A process cannot belong into a said process group if the host group values differ

Process Groups in Practice With Host Groups



Host Groups best practice

- All hosts should belong to a host group if it's dedicated hardware. Host Groups should be simple, but effective in segmenting entities.
 - Examples:
 - HOST_GROUP=Production
 - HOST_GROUP= Your app name
- In some cases Host Groups are not appropriate:
 - Kubernetes Nodes
 - Shared Infrastructure (2 apps on one host)
- Good Host Group practices eliminate a lot of process group configurations that may have otherwise be needed.

The screenshot shows the Dynatrace Monitoring overview page. On the left, there is a sidebar with various monitoring settings like Web and mobile monitoring, Cloud and virtualization, and Process group detection. The main area is titled 'Monitoring overview' and contains a table of monitored entities. The first column is 'Host' and the second is 'Summary'. A specific row for 'C-host-Gen-1' is highlighted with a blue border. The table also includes columns for 'Host units...' and 'Last check'. At the bottom right of the table, there is a small icon of a person with a gear.

Process Group Best practices

- Use Host Groups if possible, to help ensure proper grouping
- Review process groups as soon as oneagent deployment is complete
- Consider adding environment variables or other metadata on a per process level specifically for Dynatrace
- Rename process groups once proper configuration is confirmed

The screenshot shows the 'Add detection rule' dialog box. It has a title bar 'Add detection rule' and a sub-instruction 'Simple detection rules for deep monitored process group'. Below this, there are several input fields and dropdown menus:

- 'Use a [process property] to organize processes in'
- 'If Dynatrace detects the property [select attribute] and it contains (case sensitive)' (with a dropdown for 'contains')
- 'then use the value of the property [select attribute]' (with a dropdown for 'use value')
- 'in between [From delimiter] and [End delimiter]' (with dropdowns for 'from' and 'to')
- 'to [extract the identifier for the process g...]' (with a dropdown for 'extract identifier')
- 'optionally extract node identifier from [select property]' (with a dropdown for 'select property')
- 'in between [From delimiter] and [End delimiter]' (with dropdowns for 'from' and 'to')

At the bottom, there is a note: 'This feature requires OneAgent version 1.123 or higher.' and a checkbox 'Restrict this rule to processes that can be detected'. There are 'Save' and 'Cancel' buttons at the very bottom.

Tagging

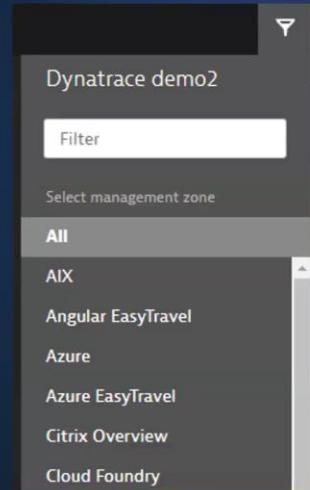
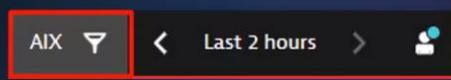
- Entity properties and/or custom metadata can be used to label entities in your environment.
- Can be used for filtering, charting, creating management zones, creating alerting profiles, and accessing information via Dynatrace API
 - It's recommended to use tags when setting up charts, management zones, etc. instead of a direct reference like properties
- **Manual tags:** Add tags ad hoc to an entity. Can be done via Settings > Tags > Manually applied tags or by clicking "+ Add tag" when viewing the entity in the Dynatrace UI.
- **Automatically applied tags (recommended):** Much more practical in a large, dynamic environment.
 - In Settings > Tags > Automatically applied tags, create a rule that determines what should be tagged and with what value
 - All entities that match the rule will be tagged automatically, including all new entities that are added to the environment

Management zones

- Powerful information-partitioning mechanism that promotes collaboration while ensuring secure access controls
- Each management zone is a set of monitored entities in your environment that are logically separated and grouped.
 - Management zones can be organized by team, application, business unit, environment, etc.
 - Management zones can overlap, just as team member responsibilities can overlap
- Once management zones are defined, users can be added to one or many management zones with varying levels of access
 - Ex: User belongs to management zone A with read only access *and* management zone B with the ability to change settings

Management zones (continued)

- Defining management zones can be done in **Settings > Preferences > Management zones**
- Create a rule that specifies which entities belong to the zone
 - Example: Services on hosts where Host group equals 'AIX' and Hosts where Host tags equal 'AIX Prod'
 - To check if the rules meet your requirements, click 'Preview' to see all entities that will be added to this zone
 - *Note: a maximum of 15 rules can be defined per management zone definition
- Once defined, you can use management zones as global filters in your environment



Let's review the session today!

- Process groups are a logical grouping based on some inherit or unique process property
 - All Process belong to a Process Group
 - All processes in a process group must all do the same function
 - A process cannot belong into a said process group if the host group values differ
- Host groups are a logical grouping and segmentation of hosts that serve the same function or grouping to fit your needs
 - A host can only apply to only one host group
 - Host groups can be by app, OS, function depending on your use case
 - Host Groups Influence Process Groups
- Adding Host groups helps with proper process grouping configuration.



Let's review!

- Tagging is used to organize your Dynatrace environment to make more sense to the users
 - Tagging can be defined manually or via automatic tagging rules, and can be applied to nearly any item that is monitored by Dynatrace
 - You can use tags to filter entities in the UI, build efficient custom charts, create management zones and alerting profiles, etc.
 - It is a best practice to use tags as filters rather than direct item references
- Management Zones are logical groups of related items.
 - Management zones work best when defined by application, organizational team, or business function
 - Used as global filter in Dynatrace environment
 - Provides secure access controls to users

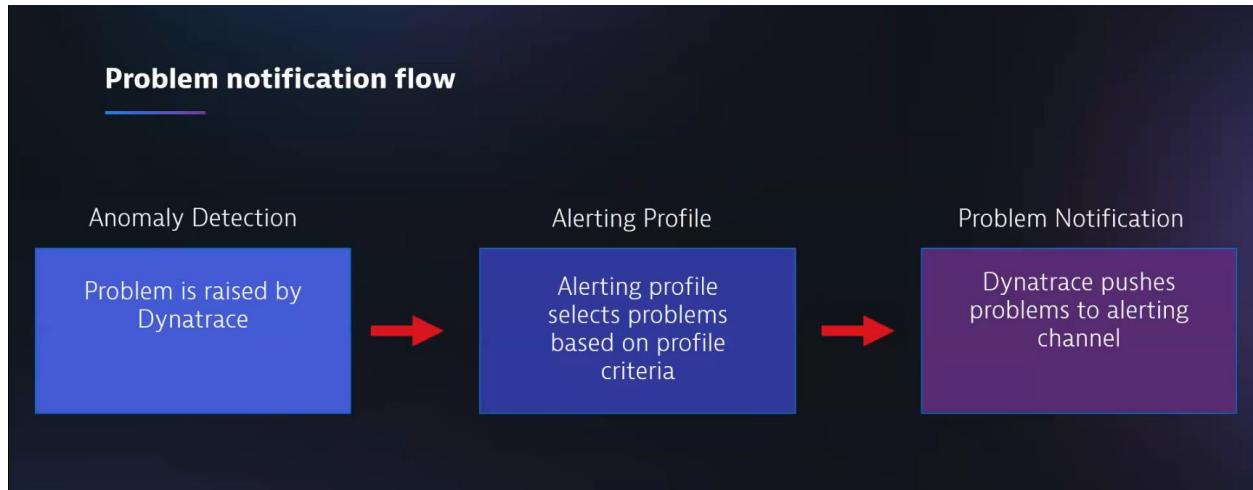
The screenshot shows the Dynatrace web interface. On the left is a dark sidebar with navigation links like Favorites, Dashboards, Problems, Observe and explore (with sub-links for Dashboards, Data explorer, Metrics, Problems, Smartscape topology, Reports), Infrastructure (with sub-links for Kubernetes, Cloud Foundry, AWS, Azure, GCP, VMware, Containers, Docker, Hosts, Technologies and processes, Host networking, Extensions, and Cloud Automation), and Cloud Automation. The main content area has a yellow header bar with a search bar, a warning message about working in a customer production environment, and buttons for 'Last 2 hours' and 'Finish session and return to Account page'. Below this is a blue navigation bar with 'Technologies' and 'Process group agent'. The main content area features a large central panel titled 'agent' with a sub-section 'Properties and tags'. This panel displays metrics for a process group: 0 Services, 1 calling Process group (Go (2)), 0 Retransmissions (0%), and 0.07% CPU usage, all running on 2 hosts. Below this is a section titled '2 Process group instances' with tabs for System performance, Networking, Technology-specific metrics, and Logs. The 'System performance' tab is selected, showing a chart for CPU usage.

Alerting profiles

- Control the delivery of problem notifications across your organization's alerting channels based on criteria in alerting profile definition
- Control exactly what conditions result in a problem notification and which don't
 - Send notifications to third-party messaging systems like Slack, OpsGenie, email, or even a custom webhook
- Creating an alerting profile can be done in **Settings > Alerting > Alerting Profiles**
 - (Optional) Define a management zone filter to focus on just the parts coming from a specific zone
 - Define severity rules (up to 20 per profile). Choose a severity level, delay the notification, and/or filter problems by tags
 - (Optional) Define event filters to further filter down what problem belong to the profile

Problem notifications

- Dynatrace has several out-of-the-box integrations with problem notification channels and test/deployment frameworks
 - Slack, OpsGenie, VictorOps, PagerDuty, JIRA, Microsoft Teams, ServiceNow, Ansible Tower, etc.
- Automatically push problem notifications to integrated alerting channel
- Open problems are continuously updated based on evolving impact and correlating events. Notifications are only pushed to third-party systems when problems are initially detected and when they are resolved.
 - Can delay notifications to external channels in alerting profile definition
- Specify an alerting profile for each problem notification integration to determine which problems get sent to which channels



The screenshot shows the Dynatrace web interface with the following details:

- Left sidebar:** Shows navigation categories like Applications & Microservices, Application Security, Digital Experience, Business Analytics, Manage, and Dynatrace Hub.
- Header:** Displays the Dynatrace logo, a search bar with the placeholder "Search CUNA Mutual Group: PROD...", and a notification badge with "0 9".
- Current Page:** "Settings > Alerting > Problem alerting profiles".
- Severity rules:** A section for defining severity rules, stating "Define severity rules for profile. A maximum of 100 severity rules is allowed." It lists four alerts:
 - Availability alert (After 10 mins; Only include entities that have all tags DLX Environment:X01)
 - Error alert (After 10 mins; Only include entities that have all tags DLX Environment:X01)
 - Resource alert (After 10 mins; Only include entities that have all tags DLX Environment:X01)
 - Slowdown alert (After 10 mins; Only include entities that have all tags DLX Environment:X01)
- Event filters:** A section for defining event filters, stating "Define event filters for profile. A maximum of 100 event filters is allowed." It shows "No data to display".
- Matches problems where:** A section showing a query: "Management zone is 'DLX - PROD'" followed by "AND" and "Error alert (After 10 mins; Only include entities that have all tags DLX Environment:X01)".
- Bottom right:** A timestamp "25:23".

dynatrace

Search CUNA Mutual Group: PROD..

You're working in a customer production environment! All your activities in this environment will be logged and visible to the customer.

Finish session and return to Account page

Applications & Microservices

- Frontend
- Services
- Kubernetes workloads
- Databases
- Queues
- Distributed traces
- Multidimensional analysis
- Profiling and optimization
- Synthetic

Application Security

Digital Experience

Business Analytics

Manage

- Dynatrace Hub
- Deployment status
- OneAgent health
- System notifications
- Access tokens
- Credential vault
- Consumption
- Settings

Integration

Configure dashboard settings

Metrics

Configure metrics settings

Integration

Integrate Dynatrace with 3rd party systems

Problem notifications

Configure dashboard settings

URL

Available placeholders

{**ImpactedEntity**}: A short description of the problem and impacted entity (or multiple impacted entities).

{**ImpactedEntityNames**}: The entity impacted by the problem.

{**NamesOfImpactedEntities**}: The names of all entities that are impacted by the problem.

{**PID**}: Unique system identifier of the reported problem.

{**ProblemDetailsText**}: All problem event details including root cause as a text-formatted string.

{**ProblemID**}: Display number of the reported problem.

{**ProblemImpact**}: Impact level of the problem. Possible values are APPLICATION, SERVICE, or INFRASTRUCTURE.

{**ProblemSeverity**}: Severity level of the problem. Possible values are AVAILABILITY, ERROR, PERFORMANCE, RESOURCE_CONTENTION, or CUSTOM_ALERT.

{**ProblemTitle**}: Short description of the problem.

{**ProblemURL**}: URL of the problem within Dynatrace.

{**State**}: Problem state. Possible values are OPEN or RESOLVED.

{**Tags**}: Comma separated list of tags that are defined for all impacted entities. To refer to the value of a specific tag, specify the tag's key in square brackets: {**Tags[key]**}. If the tag does not have any assigned value, the placeholder will be replaced by an empty string. The

Settings > **Alerting** > **Problem alerting profiles**

Integration

Integrate Dynatrace with 3rd party systems

Topology model

Configure a domain-specific topology model

Tags

Group entities using custom tags

Maintenance windows

Configure maintenance windows

Updates

Preferences

Environment settings

Consumption

Control consumption

Mainframe

Configure mainframe settings

Internal

Internal settings accessible for DevOps

Event filters

Define event filters for profile. A maximum of 100 event filters is allowed.

Add event filter

Summary	Delete	Details
Custom: Title contains 'Test Host'	NEW X	V
Predefined: Not contains events of type 'Threads resources exhausted'	NEW X	V

Matches problems where

Management zone is 'temp'

AND

Monitoring unavailable alert (After 0 mins; Include all entities)

OR

Availability alert (After 20 mins; Include all entities)

AND

Custom: Title contains 'Test Host'

AND

Predefined: Not contains events of type 'Threads resources exhausted'

Settings > Alerting > Problem alerting profiles

Configure a domain-specific topology model

Add event filter

	Summary	Delete	Details
Tags	Custom: Title contains 'Test Host'	<small>NEW</small>	X
Maintenance windows	Predefined: Not contains events of type 'Threads resources exhausted'	<small>NEW</small>	X
Updates			
Preferences	Management zone is 'temp'		
Consumption	AND Monitoring unavailable alert (After 0 mins; Include all entities) OR Availability alert (After 20 mins; Include all entities)		
Mainframe	AND Custom: Title contains 'Test Host'		
Internal	AND Predefined: Not contains events of type 'Threads resources exhausted'		
	dddd	X	
	Default		
	Default for ActiveGate Token Sync		

Dynatrace 2024 Videos Content



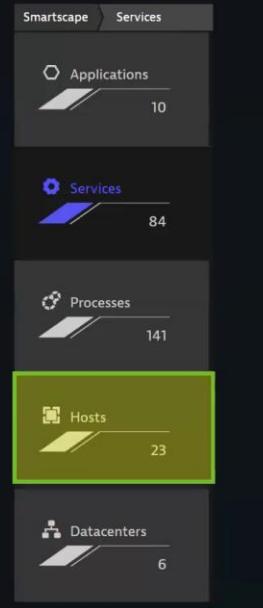
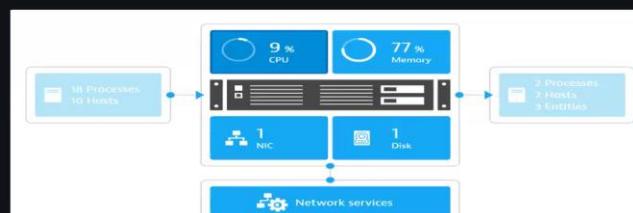
The Datacenter Layer

- A physical location on Earth where hosts reside.
- An organization can have one or multiple datacenters in use
- Related terms: vCenter, zone, Virtual, AWS, Geo Region



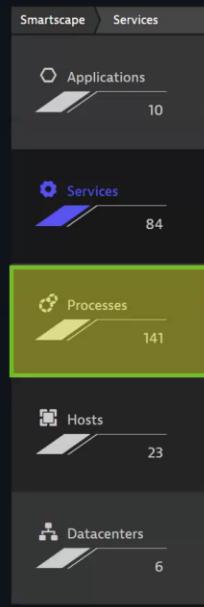
The Host Layer

- A physical or virtualized operating system
- The source of compute, memory, and storage resources
- Host related metrics reside here including:
 - CPU
 - Memory (RAM)
 - Network interface Cards (NICs)
 - Disk Storage and Utilization



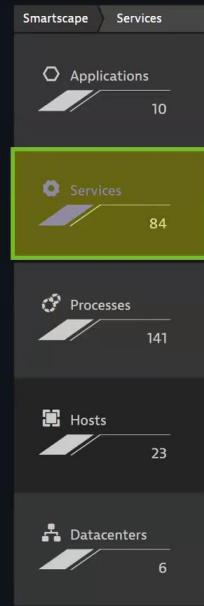
The Process Layer

- A currently executing computer program on a host
 - Think Java, .NET, Jetty Processes
- A means for code to request computing resources
- Metrics include:
 - Process CPU
 - JVM Heap
 - PID
 - Threads
- Processes are both tangibly and logically grouped in Dynatrace.
 - We have 3 Java application JVMs across 3 hosts = Tangible=Process group Instance
 - Those 3 Java JVMs make up my Prod Application server= Logical= Process Groups



The Service Layer

- A set of code or classes that performs some type of business logic
- Services are logical groupings that run on a Process or Processes
- Transactions occur at this layer
 - Web requests
 - Web services
 - Database requests
- Metrics include:
 - Failure rate
 - Response time
 - Throughput
 - Exceptions



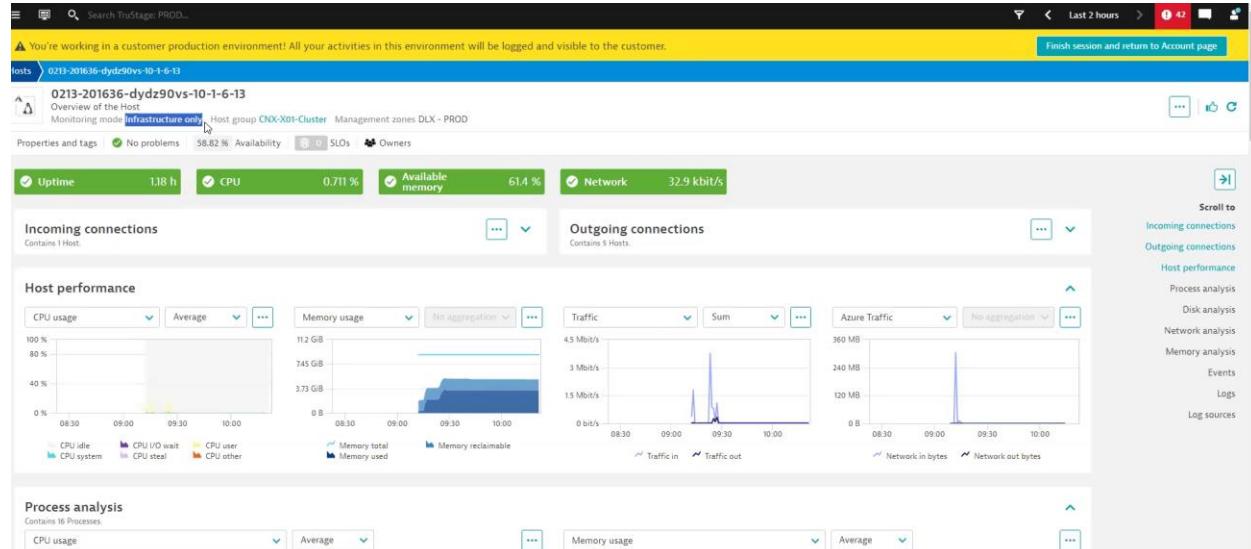
The Application Layer

- The point that End users interact with Services to perform business logic.
- Think User analytics
- Metrics in this layer include:
 - Browser
 - User action time
 - W3c timings
 - Conversion goals
 - Location distribution
- Applications are logical Groupings based on URL or Domain



We have two types of monitoring

1. Full stack monitoring (It will show all the details, like services, syntactic monitoring, host etc.)
2. Infrastructure monitoring (It will show CPU, memory and disk usage)



Search TruStage PROD...

You're working in a customer production environment! All your activities in this environment will be logged and visible to the customer. Finish session and return to Account page

Hosts > VAZSPWSDM2P02.CMUTUAL.com > Processes > IIS app pool 945d990a958347deb3df654dd8f41865

ASP.NET IIS app pool 945d990a958347deb3df654dd8f41865

Properties and tags | ChernwellCategory: Preplanning | Environment: Prod | ManagementZone: Preplanning | Service: 4fa04e87ce034...

No problems Today, 08:36 - 10:36
100% Availability in last 7 days
0 min total downtime Last downtime on today, 00:02

20. Aug 21. Aug 22. Aug 23. Aug 24. Aug 25. Aug 26. Aug
Running Shutdown

No events Today, 08:36 - 10:36 | Process crash details | Go to logs

No logs available | Go to logs

Provided services
Web request services: /4fa04e87ce034373a94d10e802716c1d on SharePoint Web Services
/45415018f0145178626572043bd655 on SharePoint Web Services
Web services: BIMonitoringServiceApplication
ProfileDBCacheService

View detailed CPU breakdown

You're working in a customer production environment! All your activities in this environment will be logged and visible to the customer. Finish session and return to Account page

Services > /45a04e87ce034373a94d10e802716c1d on SharePoint Web Services

ASP.NET Overview of the Service

Properties and tags | No problems | SLOs | Owners

Service overview

Response time | Failed requests | Throughput

Key requests/endpoints
Contains 0 Requests

Name | Response time [Median] | Failure rate [Average] | Failures [Average] | Throughput | Actions | Details

No key requests/endpoints were found in the selected timeframe
Go to service details, select a request and mark it as key request.

View all request details | Top web requests | View outliers

Scroll to Service overview
Key requests/endpoints
Topology
Distributed traces
Events
Related logs

Search TruStage PROD...

You're working in a customer production environment! All your activities in this environment will be logged and visible to the customer.

Last 2 hours

Finish session and return to Account page

Services /45a04e87ce034373a94d10e802716c1d on SharePoint Web Services

ASP.NET Overview of the Service

Properties and tags No problems SLOs Owners

Response time (p50) Response time (p99) Response time (p999) Failure rate Failures Throughput

Key requests/endpoints Contains 0 Requests

No key requests/endpoints were found in the selected timeframe. Go to service details, select a request and mark it as key request.

View all request details Top web requests View outliers

Scroll to Service overview Key requests/endpoints

Topology Distributed traces Events Related logs

Topology

Related services (0) Processes and hosts (2)

View backtrace Service flow

Related services Contains 0 Services Metrics are overall metrics and not limited to calls from or to this service.

Name	Relation	Response time	Throughput	Failure rate	Details
No related services	There were no related services (incoming nor outgoing) in the selected timeframe.				

Applications

271 Applications

Monitoring settings

Search

All 271

Status Monitored 265 Unmonitored 6

Application type Web applications 271 Mobile apps 0 Custom applications 0

Injection type Auto injected 271

Tag Advisor Workspace 1 BFY-PROD 1 CherwellCategory 104

Name	Apdex	Actions	Load actions	XHR actions	Errors	3rd party / CDN
www.trustage.com JavaScript error rate increase	Good	325 /min	Visually complete: 2.4 s	Visually complete: 0.4 s	913 /min	6.38k /min
Digital Rep Tools	Unacceptable	0.29 /min	Visually complete: 2.1 s	-	0.62 /min	0.00 /min
rapidonline.trustage.com	Unacceptable	2.07 /min	Visually complete: 0.9 s	-	8.08 /min	3.55 /min
wholesale.trustage.com	Unacceptable	1.99 /min	Visually complete: 0.8 s	-	2.04 /min	26 /min
SalesSuppRpt	Unacceptable	3.3 /min	Visually complete: 1.3 s	-	8.3 /min	0.00 /min
advisorworkspace.trustage.com	Unacceptable	3.02 /min	Visually complete: 2.2 s	-	39 /min	70.4 /min
billpay.trustage.com	Unacceptable	1.36 /min	Visually complete: 1.4 s	-	8.11 /min	4.07 /min
insidewipspciweb	Unacceptable	0.35 /min	Visually complete: 5.3 s	-	3.97 /min	0.00 /min
lendingclaim.trustage.com	Unacceptable	3.31 /min	Visually complete: 1.5 s	-	3.25 /min	72.8 /min
dthchawebsvcs.mutual.com	Unacceptable	0.50 /min	Visually complete: 1.5 s	-	1.8 /min	0.06 /min
claims.trustage.com	Unacceptable	31.8 /min	Visually complete: 0.7 s	Visually complete: 0.1 s	41.7 /min	139 /min
cunatpppproebix.com	Unacceptable	14.3 /min	Visually complete: 2.1 s	-	18.9 /min	0.00 /min

Applications www.trustage.com

Show Median metrics for XHR actions

Median Total XHR actions / min XHR actions / min with visually complete

Compare to previous time frame Analyze performance

Top 3 user actions

Top actions based on total Time consumed (Duration and Actions/min)

Action	Time consumed	Actions
loading of page /life-insurance/quote	18.13 /min	Actions 18.13 /min
loading of page /	16.00 /min	Actions 16.00 /min
loading of page /my-account/login	13.87 /min	Actions 13.87 /min

View full details

Composite metrics across response times

Median metrics of Load actions

3.23s User action duration 2.39s Visually complete 1.77s Speed Index 2.01s DOM interactive 2.60s Load event end

Show all metrics

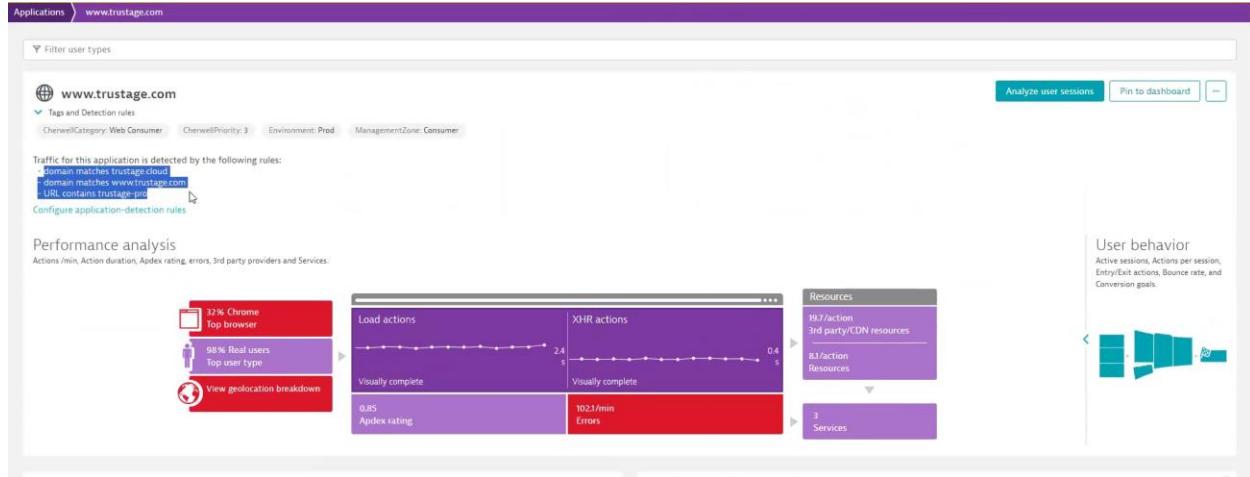
Compare to previous time frame Analyze performance

Top errors

A change in error levels may indicate a device- or browser-specific problem.

Requests (786 /min) JavaScript (89.67 /min) Custom

Top 3 pages



Service: flow , backtrace & traces

What are we covering Today?

- Understand the Service Layer of the Smartscape and it's key metrics
- Find out what's a Service in depth
- Explore some of the UI components of the Service layer
- Review the following UI menus
 - Service overview
 - Service Flow
 - Service Backtrace
 - Distributed Traces



The Service Layer

- A set of code or classes that performs some type of business logic
- Services are logical groupings that run on a Process or Processes
- Transactions occur at this layer
 - Web Requests
 - Web Services
 - Database requests
- Metrics include:
 - Response Time
 - Failure Rate
 - Throughput
 - Exceptions



What are Services In Practice?

- Services are where transactions occur for your application
 - Example: Your travel website has functions/requests (add to cart, sign in, search etc.)
 - The Customer front end service contains these functions or requests **within the service**
- Services often call other services to get the task or request completed.
 - Search requests usually call a database service
- Services usually fall under common types:
 - Web requests
 - Web services
 - Database Calls
 - Message Queues
 - A custom entry point

What do we get with Service monitoring?

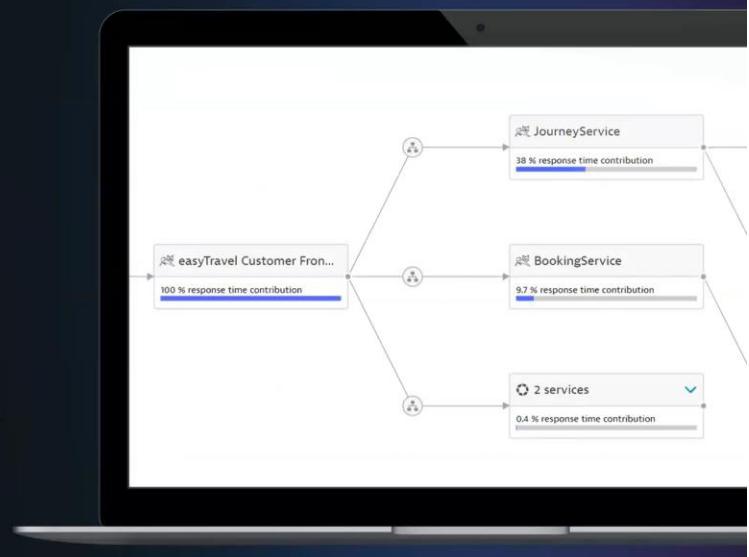
- Dynatrace monitoring of services extends all the way down to the monitoring of discrete methods (code) and code level information is collected for each request, known as Purepaths
- Every Service (and transaction within) has a dynamic baseline calculated that considers Response time and Failure rate which is "normal" for your service in question.
- Services can be renamed, and merged to fit your use cases.

The screenshot shows a mobile device displaying the Dynatrace interface. The top bar is blue with the text 'Top requests'. Below it is a search bar with the placeholder 'Start typing to filter'. A table lists several transactions:

Name	Icefaces	Client IP Ad...	Content-Len...
/special-offers.jsp	Icefaces	Client IP Ad...	Content-Len...
/CalculateRecommendations	Icefaces	Client IP Ad...	Content-Len...
/orange.jsf	Icefaces	Client IP Ad...	Content-Len...
easyTravelID...	easyTravel D...		
/legal-orange-mobile.jsf	Icefaces	Client IP Ad...	

Device frame - laptop

- Services are usually automatically detected, they include:
 - Web services
 - Web requests,
 - Database requests
 - Message queues
 - Custom services
- Services may call one or many other services to complete their transactions



Services > easyTravel Customer Frontend

easyTravel Customer Frontend

Seen recently

Properties and tags

1 Application, 0 Service, 0 Network clients

5.47k/min Throughput, 1 Apache to...

4 Services, 0 Database

Smartscape view

Dynamic web requests

Response time: 4.26 ms, Failure rate: 0%, Throughput: 673/min

CPU: 5.48 ms/req, Throughput: 673/min

[View dynamic requests](#)

Resource requests

0.06 ms, 0 %, 0.21 ms/req, 4.79k/min

3 Problems in last 72 hours

Current hotspots

- High consumption of service resources 33% /orange-booking-payment.jft
- Slow response time 242 ms /CalculateRecommendations

Understand dependencies Today, 09:03 - 11:03

- Understand all dependencies and response time contributions [View service flow](#)
- Understand which user actions and related services are dependent on this service [Analyze backtrace](#)

No events Today, 09:03 - 11:03

Services > easyTravel Customer Frontend

easyTravel Customer Frontend

Seen recently

Properties and tags

1 Application, 0 Service, 0 Network clients

5.47k/min Throughput, 1 Apache to...

4 Services, 0 Database

Smartscape view

Infographic and Service overview

5.47k/min Throughput, 1 Apache to...

Dynamic web requests

Response time: 4.26 ms, Failure rate: 0%, Throughput: 673/min

CPU: 5.48 ms/req, Throughput: 673/min

[View dynamic requests](#)

Resource requests

0.06 ms, 0 %, 0.21 ms/req, 4.79k/min

Problem history

3 Problems in last 72 hours

Hotspot Analysis

Current hotspots

- High consumption of service resources 33% /orange-booking-payment.jft
- Slow response time 242 ms /CalculateRecommendations

Drilldown into Dependencies

Understand dependencies Today, 09:03 - 11:03

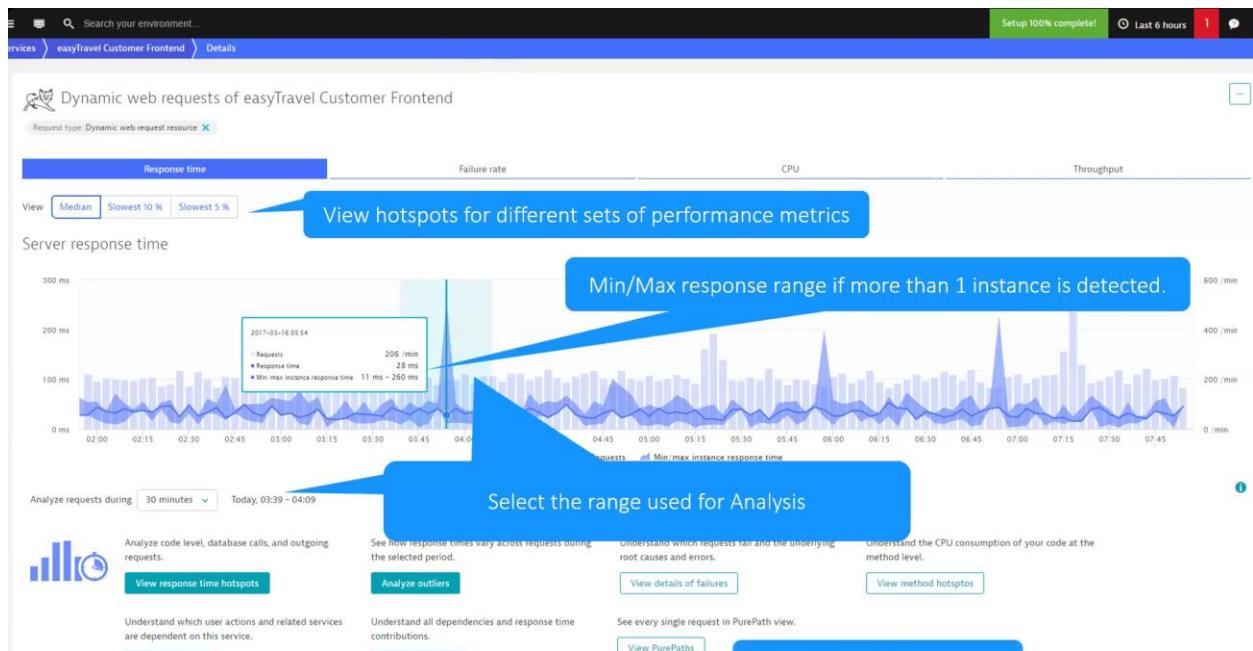
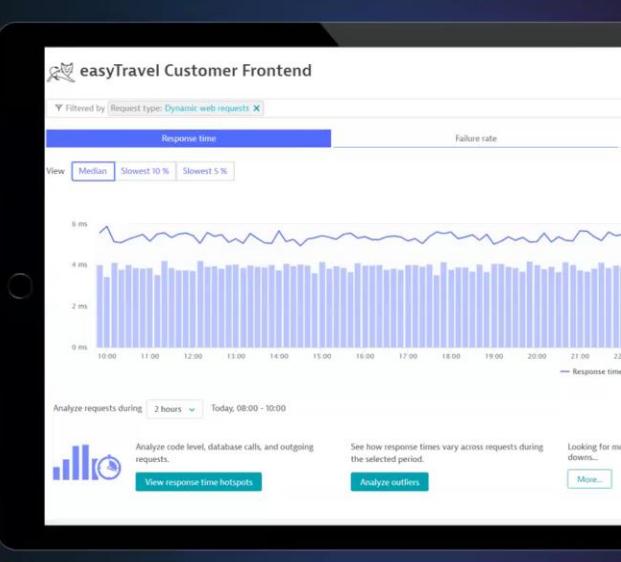
- Understand all dependencies and response time contributions [View service flow](#)
- Understand which user actions and related services are dependent on this service [Analyze backtrace](#)

Eventlogger

No events Today, 09:03 - 11:03

Service Details

- What is it?
 - Detailed overview of a service's performance (requests, throughput, failures)
 - Starting point for further analysis
- When would I use it?
 - Find Requests that exists within a given service
 - Find response time, failure rate, throughput for a given request.



Name	Total time consumption	Median response time ▾
com.dynatrace.easytravel.customer.frontend.jar easytravel-2.0.0-x64 gdn-rx-ose12-acc-e01v.emea.cpwr.corp	248 ms	
com.dynatrace.easytravel.customer.frontend.jar easytravel-2.0.0-x64 gdn-rx-ose12-acc-e102v.emea.cpwr.corp	22.9 ms	
com.dynatrace.easytravel.customer.frontend.jar easytravel-2.0.0-x64 gdn-rx-rh6-acc-e01v.emea.cpwr.corp	17 ms	
com.dynatrace.easytravel.customer.frontend.jar easytravel-2.0.0-x64 gdn-rx-co6-acc-e02v.emea.cpwr.corp	2 ms	

Key requests

Add key requests by opening a request from the top requests list and mark them as key request. Why? [Read more...](#)

A 'Key Request' can be marked manually, which keeps data long term

Top web requests Showing last values [Change web request naming rules](#)

Name	Total time consumption ▾	Median response time
/orange.jsf		180 ms
/orange-booking-finish.jsf		20.7 ms
/orange-booking-payment.jsf		59.5 ms
/special-offers.jsp		1 s
/logout.jsf		195 ms
/orange-booking-review.jsf		51.4 ms
/services/JourneyService/findLocations		184 ms
/orange-trip-details.jsf		81.9 ms
/contact-orange.jsf		19.2 ms
/services/JourneyService/FindJourneys		96.9 ms

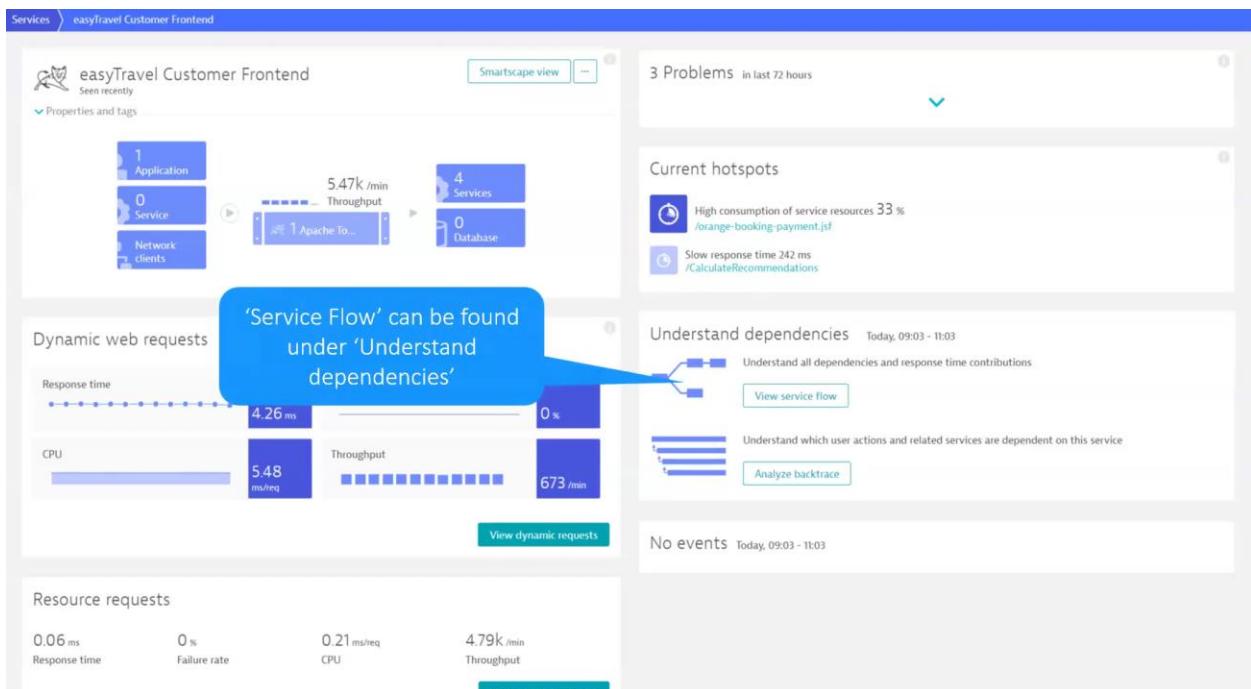
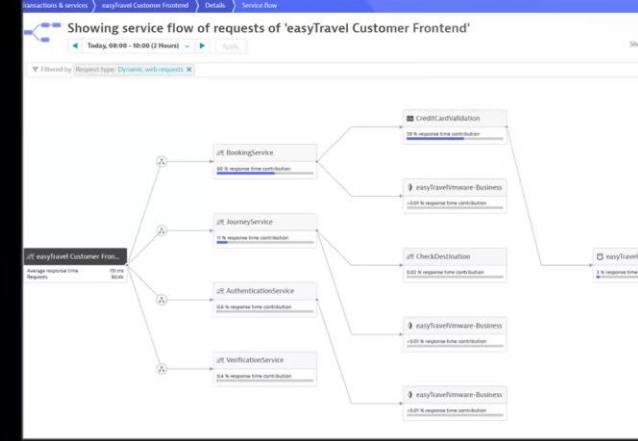
All requests executed in the selected timeframe

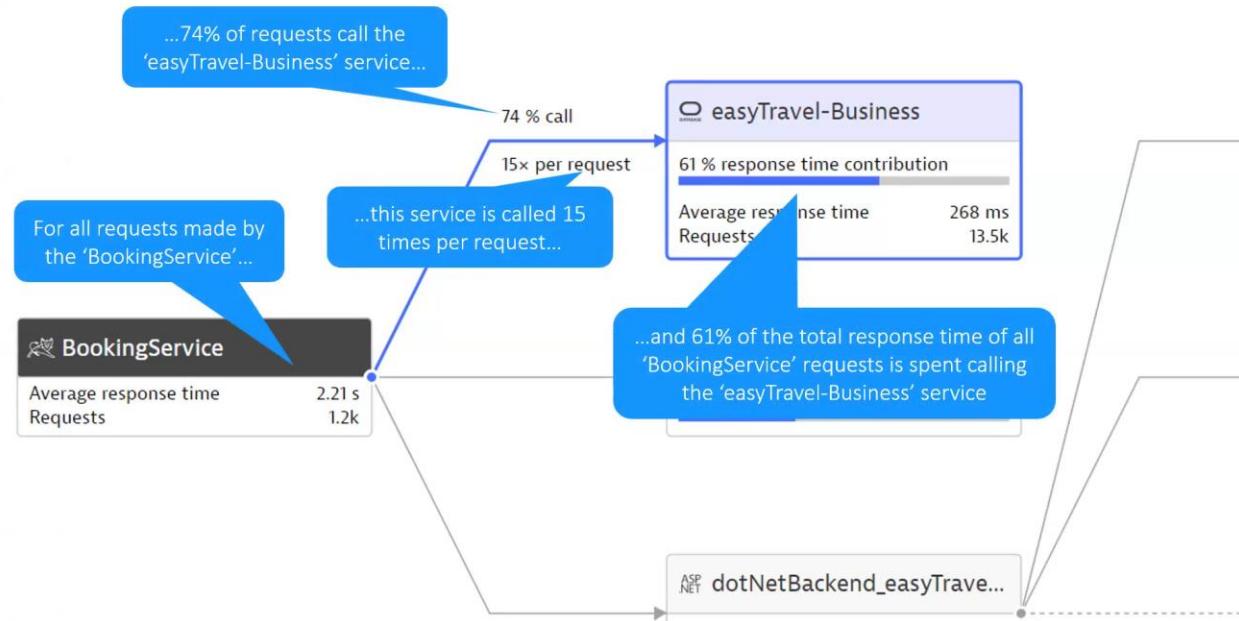
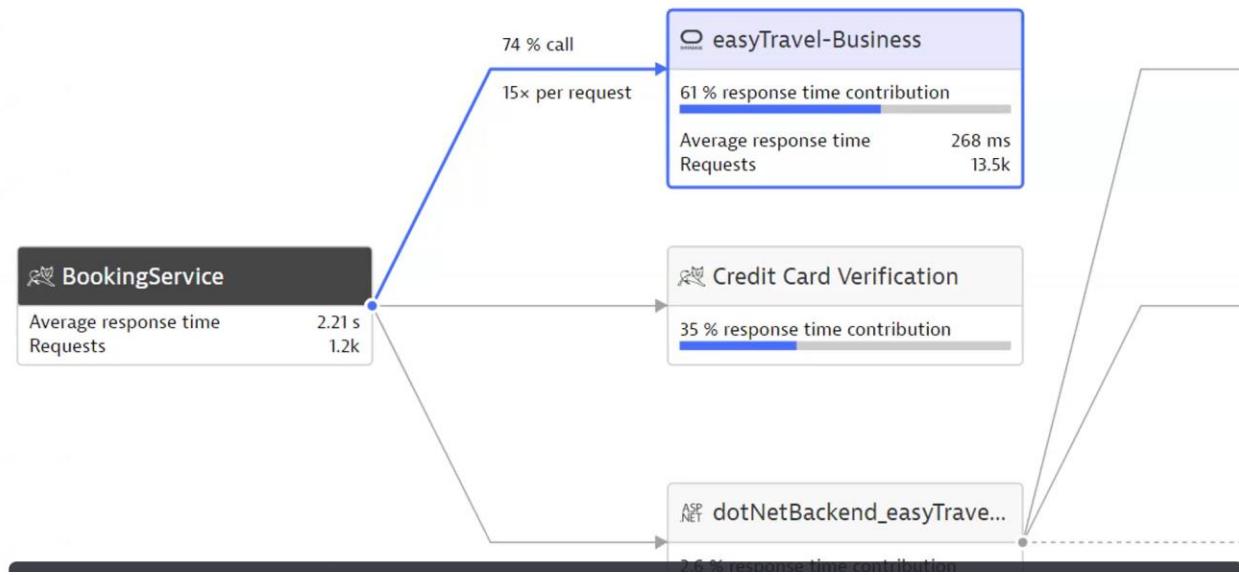
Drilldown to view Details for a specific request

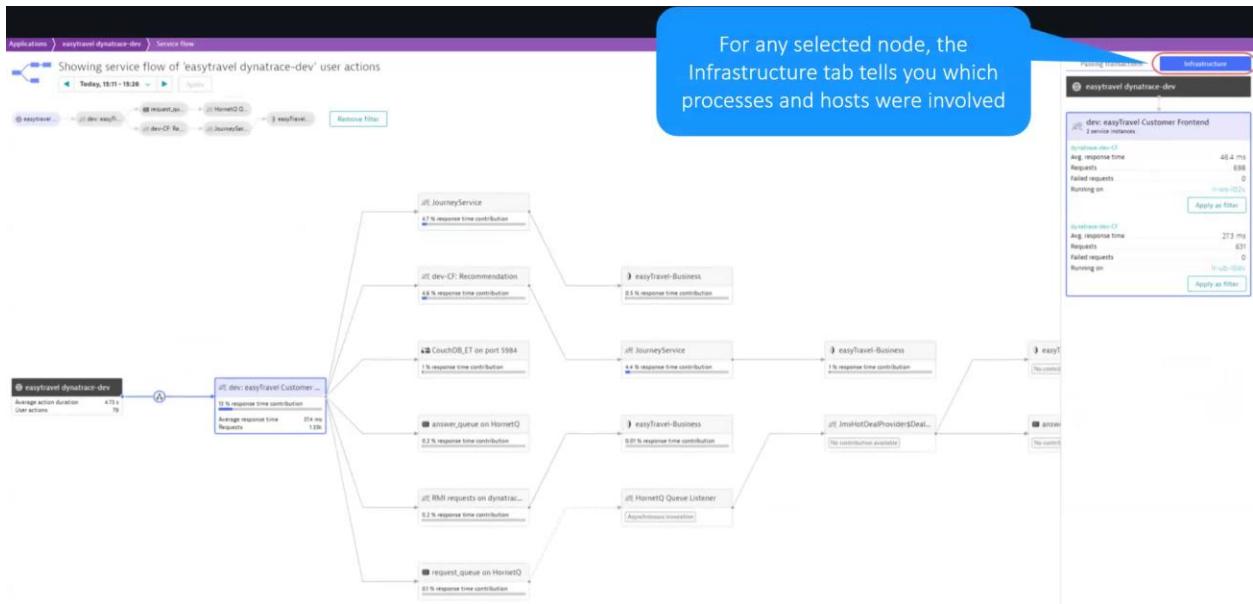
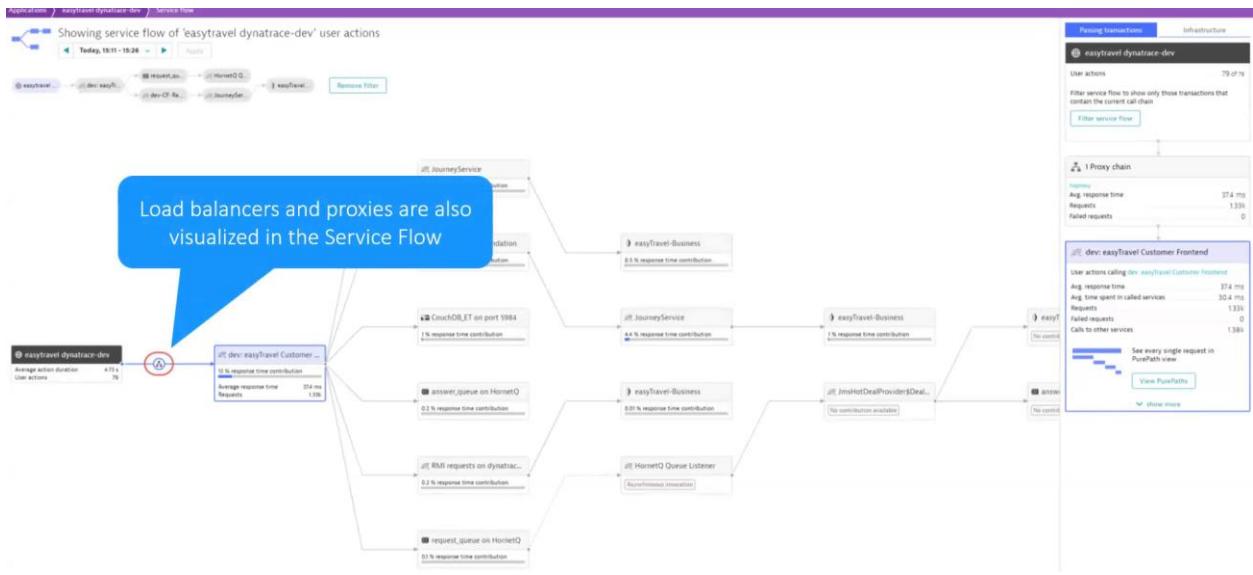


Service Analysis – Service Flow

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







Service Analysis – Service Backtrace

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

The screenshot shows a mobile device displaying the 'Backtrace' section of the service details for 'CouchDB_ET on port 5984'. At the top, there's a header bar with 'Services > CouchDB_ET on port 5984 > Details > Backtrace'. Below the header, a title says 'Service-level backtrace of requests to 'CouchDB_ET' today, 13:38 - 15:38' with a 'Add filter' button. A descriptive text explains that the tree view represents the sequence of calls triggered by user actions. The main area is titled 'Incoming requests to this service' and lists several services and applications that make calls to the selected service:

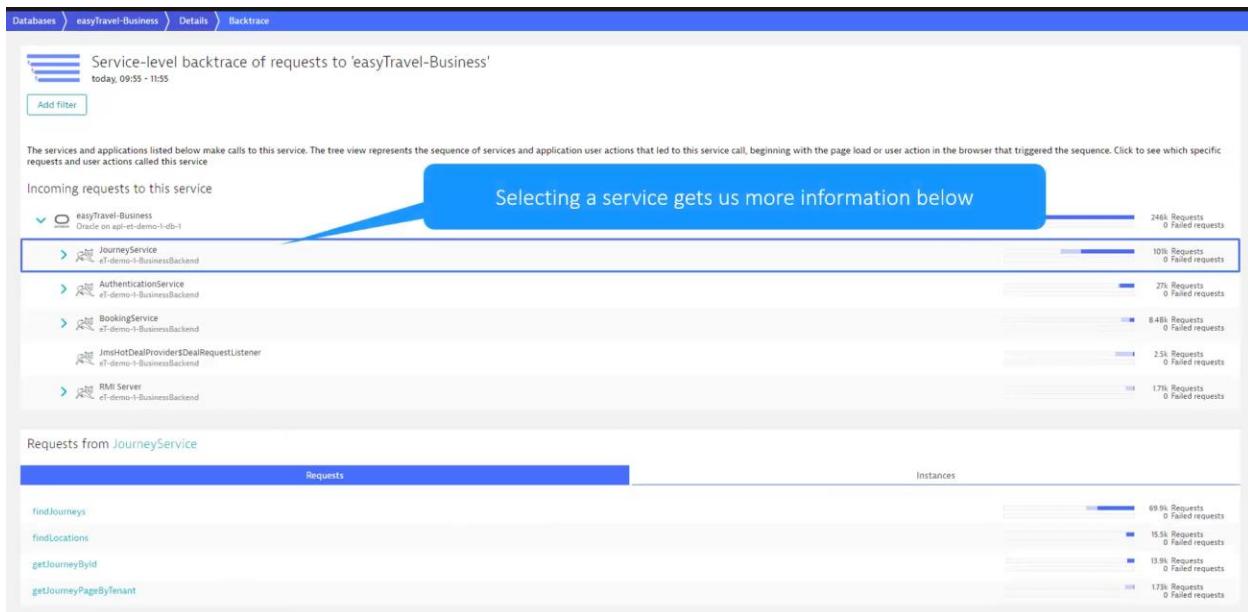
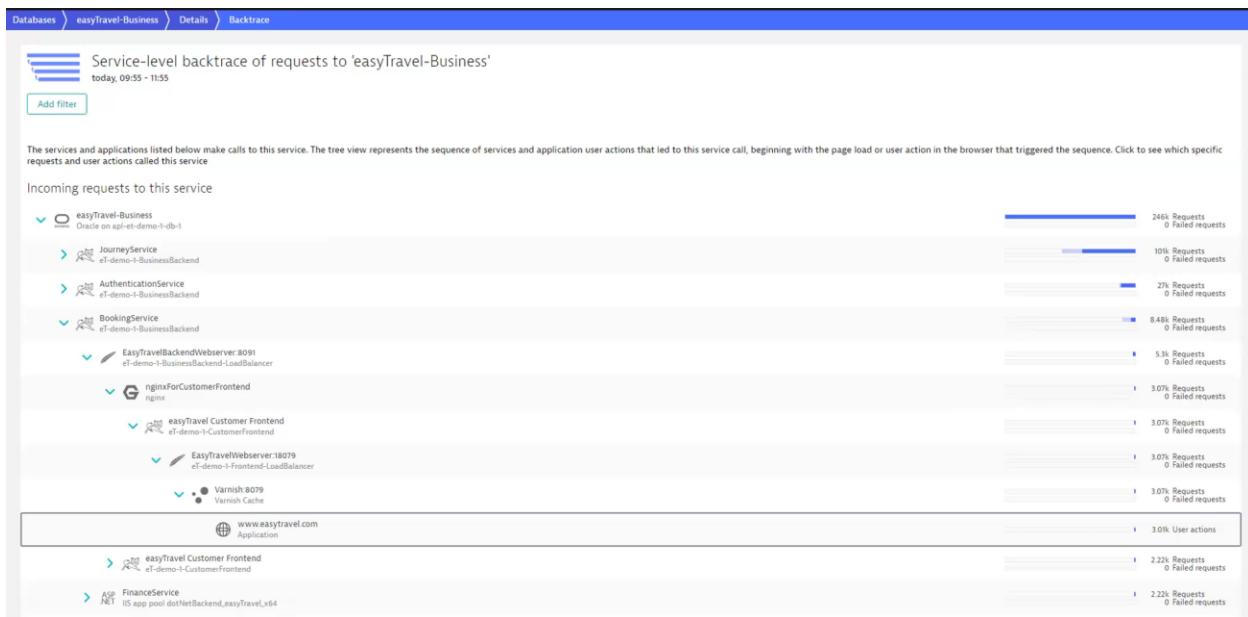
- CouchDB_ET on port 5984
- easyTravel Customer Frontend (el-demo-1-CustomerFrontend)
- Varnish:8079 (Varnish Cache)
- www.weather.easytravel.com (Application)
- www.easytravel.com (Application)

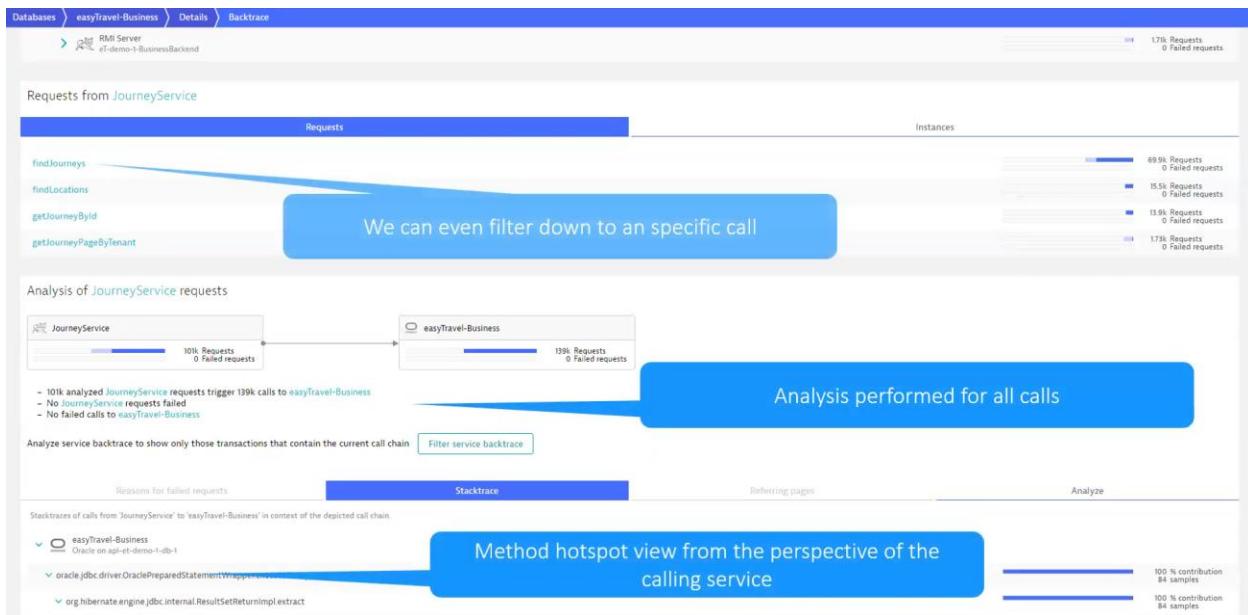
The screenshot shows a mobile device displaying the 'Backtrace' section of the database details for 'easyTravel-Business'. At the top, there's a header bar with 'Databases > easyTravel-Business > Details > Backtrace'. Below the header, a title says 'Service-level backtrace of requests to 'easyTravel-Business'' today, 09:55 - 11:55' with a 'Add filter' button. A large blue callout bubble points from the top right towards the center of the screen, containing the text 'Which services call the 'easyTravel-Business' database service?'. A descriptive text explains that the tree view represents the sequence of calls triggered by user actions. The main area is titled 'Incoming requests to this service' and lists several services and applications that make calls to the selected database service:

- easyTravel-Business Oracle on apl-et-demo-1-db-1
- JourneyService (el-demo-1-BusinessBackend)
- AuthenticationService (el-demo-1-BusinessBackend)
- BookingService (el-demo-1-BusinessBackend)
- JmsHotDealProvidersDealRequestListener (el-demo-1-BusinessBackend)
- RMI Server (el-demo-1-BusinessBackend)

On the right side of the screen, there are six horizontal bars representing request statistics for different services:

- 246k Requests, 0 Failed requests
- 101k Requests, 0 Failed requests
- 27k Requests, 0 Failed requests
- 8.48k Requests, 0 Failed requests
- 2.5k Requests, 0 Failed requests
- 171k Requests, 0 Failed requests





Purepaths

- What is it?
 - A mapping of an individual request that exists within a service. All metrics on the service layer are based on purepath(s) data.
- Single request analysis
 - Deep problem triaging
 - Request attributes (custom meta data)
- Individual purepaths should be the last step in problem triage. Filters are a must.
- Every menu aggregates, interprets and presents Purepath Data
- PurePaths are now listed as Traces

The screenshot shows the Dynatrace PurePath analysis interface for a trace named 'orange-booking-review.jsf'. It displays execution breakdowns, request attributes (Client IP Address, User Agent, Journey ID), and a detailed stack trace with 29 stack frames. The stack trace shows various layers of the application, including the customer frontend, business logic, and database access.

Services ConfigurationService CloudAutomation

Properties and tags

CloudAutomation Application: EasyTravel CloudAutomation EasyTravel: BackEnd

CloudAutomation Stage: Staging

No hotspots detected Today, 15:52 - 15:57

Multidimensional analysis views [Create analysis view](#)

This section will list your bookmarked multidimensional analysis views for this service. Click 'Create analysis view' to start.

Understand dependencies Today, 15:52 - 15:57

- Understand sequence and response times of service calls of each request. [View service flow](#)
- Understand which user actions and related services depend on this service. [View backtrace](#)
- See requests in trace overview. [View distributed traces](#)
- Understand and analyze which web requests are the most expensive and most frequently called. [View requests](#)

Requests

Metric	Value
Response time	907 µs
Failure rate	0 %
CPU	987 µs/req
Throughput	40.7 /min

Services ConfigurationService CloudAutomation Details Distributed traces

Browse and find [distributed traces](#) related with this service.

zoom.us - To exit full screen, press **Esc**

Request count Response Time (Median)

Most recent 187 traces

Search for names and http methods... [Configure columns](#)

Details	Start time	Name	Response time	Processing time	Http method	Response code	Actions
Oct 03 15:57:24.905	getAllPluginNames /services/ConfigurationService/		1.27 ms	1.27 ms	POST	200	...
Oct 03 15:57:24.901	getEnabledPluginNames /services/ConfigurationService/		811 µs	811 µs	POST		
Oct 03 15:57:24.082	getEnabledPluginNames /services/ConfigurationService/		945 µs	945 µs	POST		
Oct 03 15:57:20.237	getEnabledPluginNames /services/ConfigurationService/getEnabledPluginNames		937 µs	937 µs	GET		
Oct 03 15:57:15.237	getEnabledPluginNames /services/ConfigurationService/getEnabledPluginNames		573 µs	573 µs	GET		

Analyze Oct 03 15:52 - 15:57

- Trace
- Service flow
- Service backtrace
- Details of failures

Distributed traces b7a775248c46f1928eb926b4431eee3

getEnabledPluginNames

Summary Timing Threads Code level Logs Errors

Topology

Service	ConfigurationService CloudAutomation
Request	getEnabledPluginNames
Host	easytravel.staging
Process	com.dynatrace.easytravel.business.backend.jar easytravel-*-*
Operating system	Linux - Ubuntu 20.04.2 LTS (Focal Fossa) (x86)
Bitness	64-bit
Service type	Web service
Service main technology	Apache Tomcat
Technology	Apache HTTP client (4.5.10), OneAgent SDK (Java 1.7.0), QOS Logback (1.2.3), Apache Tomcat (7.0.93.0), Java (OpenJDK 11.0.16), OpenTracing (Java 0.33.0), Apache Axis, OpenTelemetry (Java)

Request attributes

Booking	getEnabledPluginNames
ClientIP	127.0.0.1
sessionid	localhost:8091/services/ConfigurationService/getEnabledPluginNames

Metadata

URI	/services/ConfigurationService/getEnabledPluginNames
HTTP method	GET
Response status	200 - OK
Application ID	easyTravel Business Backend
Context root	/
Server name	localhost
Client IP	127.0.0.1
Web service class	com.dynatrace.easytravel.business.webservice.ConfigurationService

Distributed traces b7a775248c46f1928eb926b4431eee3

getEnabledPluginNames

Summary Timing Threads Code level Logs Errors

Search name, url, sql, attribute...

doFilter

- Built-In Spring | org.springframework.web.filter.OncePerRequestFilter

invokeBusinessLogic

- Built-In Apache | org.apache.axis2.rpc.receivers.RPCMessageReceiver

send

- Built-In Apache | org.apache.axis2.engine.AxisEngine

	Elapsed	Self time	Duration	Actions
doFilter	0 ms	245 µs	589 µs	...
invokeBusinessLogic	+202 µs	172 µs	172 µs	...
send	+381 µs	172 µs	172 µs	...

Let's review the session today!

- Understand the Service Layer of the Smartscape and it's key metrics
 - Think Transactions or business logic
 - Response time, failure rate, throughput
- Explored some of the UI components of the Service layer
- Review the following UI menus
 - Service overview
 - Service Flow
 - Downstream service to service communication
 - Service Backtrace
 - Upstream service to service communication
 - PurePaths
 - An individual request that exists within a service



This screenshot shows the Dynatrace Service Overview dashboard. The left sidebar includes sections for Infrastructure Observability (Kubernetes, Cloud Foundry, AWS, Azure, GCP, VMware, Containers, Hosts, Technologies & Processes, Host Networking, Extensions) and Application Observability (Frontend, Services, Kubernetes Workloads, Database Services, Message Queues, Distributed Traces, Multidimensional Analysis, Profiling & optimization, Synthetic). The main content area displays a "Services" list with 2.39k Services. A filter sidebar on the left allows filtering by Service type (Web service, Web request service, Custom service, Messaging service, RMI service), Problem impact (Any, Impacted, Not impacted), and Technology (.NET, .NET Remoting). The right side shows a detailed view of a service named "(IBraveServer) on Catalina/localhost" with metrics like Requests/min (11.5k/min) and Actions. A sidebar on the right indicates the current session is from "TruStage PROD" and offers options to "Finish session and return to Account page", "Update", and "Pin to dashboard".

You're working in a customer production environment! All your activities in this environment will be logged and visible to the customer. [Finish session and return to Account page](#)

Services > Sitecore Managed Services PROD A > Settings > Web request naming

Naming

Failure detection

Web request naming

Anomaly detection

Key requests

Muted requests

Generate clean URLs

Dynatrace automatically removes certain IDs from web request names for better grouping. You can change this default behavior and also add your own cleanup rules.

Remove UUIDs, IP addresses and IBANs from URLs

Create clean URL rule No clean URL rules defined yet!

In tracking this service's requests, Dynatrace tracks each URL path separately. In some instances, this approach is too granular and inadvertently makes the structuring of your requests appear to be more complex than they actually are. Request naming rules enable you to consolidate multiple related URLs into a single request for monitoring purposes. [More...](#)

Add rule No rules defined yet!

Global request naming rules

Dynatrace tracks each request separately. [More...](#)

Define global request naming rules on the [Dynatrace API](#). Global naming rules can be overridden on the service level. For more information please check the [Dynatrace Documentation](#).

You're working in a customer production environment! All your activities in this environment will be logged and visible to the customer. [Finish session and return to Account page](#)

Services > Sitecore Managed Services PROD A > Settings > Anomaly detection

Failure detection

Web request naming

Anomaly detection

Key requests

Muted requests

No settings defined on the current entity, using defaults set on Environment

Response time

Detect response time degradations

Detection mode for response time degradations

Automatic

All requests

Alert if the median response time of all requests degrades beyond **both** the absolute and relative thresholds:

Absolute threshold
100 ms

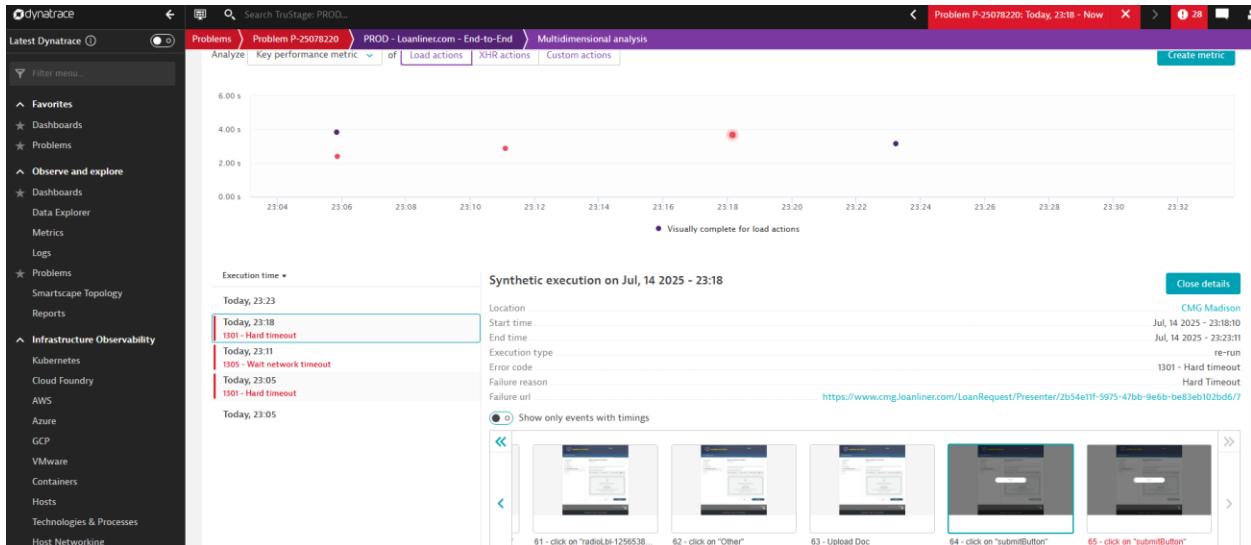
Relative threshold
50 %

Slowest 10%

Alert if the response time of the slowest 10% of requests degrades beyond **both** the absolute and relative thresholds:

Absolute threshold
1,000 ms

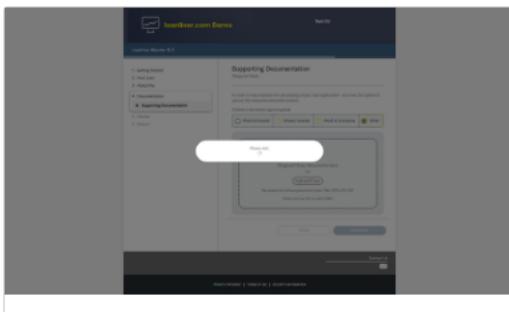
Relative threshold



Event: 64 - click on "submitButton"

Type: Click

Actual



Show difference (99.43%)

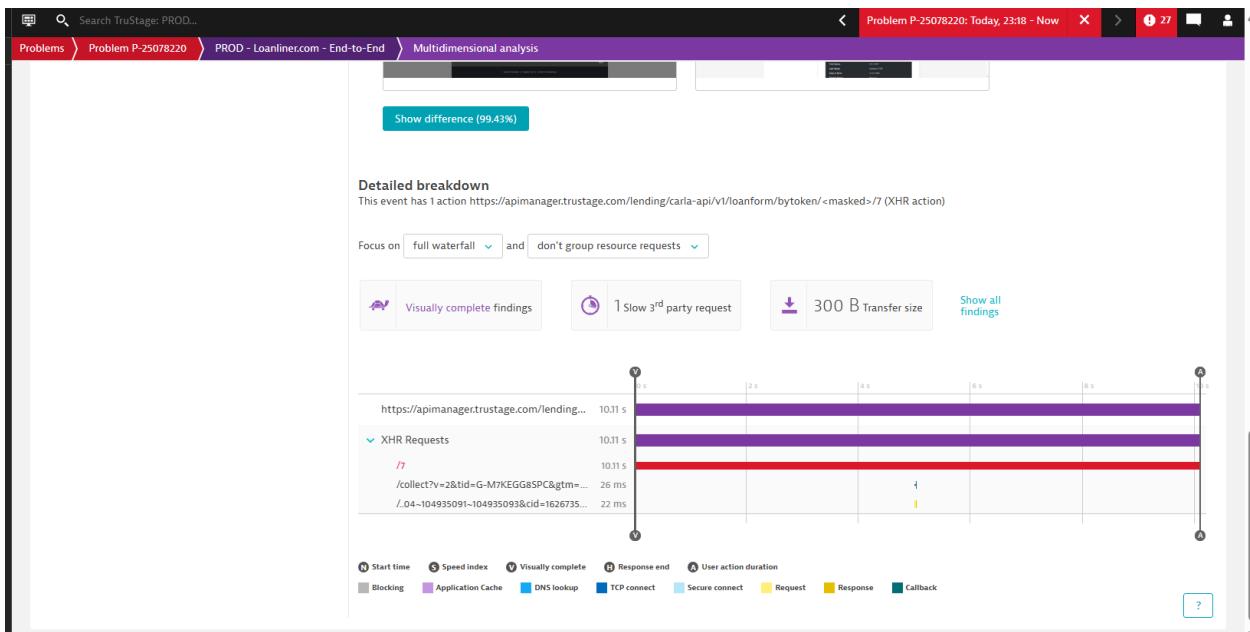
Expected (screenshot from Jul, 14 2025 - 23:09 in CMG Madison)



Detailed breakdown

This event has 1 action <https://apimanager.trustage.com/lending/carla-api/v1/loanform/bytoken/<masked>/7> (XHR action)

Focus on and



By Application

Dynatrace is an Application Performance Management (APM) monitoring system that instruments applications at the code level to collect and analyze the health of every transaction passing through the application.

Dynatrace brings infrastructure and cloud, application performance, and digital experience monitoring into an all-in-one, automated solution that's powered by artificial intelligence.

Dynatrace assists in driving performance results by providing development, operations, and business teams with a shared platform. The core benefit of Dynatrace is lowering MTTR by helping Ops, Dev and other support teams focus into the root cause of a problem without sorting through dozens of logs and different toolsets.

- **OneAgent** - Dynatrace installer for enabling full infrastructure monitoring. Without it, your server may still be detected in Dynatrace with limited monitoring functionality.
- **Host Group** - Application-level grouping to allow group level setting of tagging and process groups and other configs.
- **Management Zone** - Permits Power Users to modify configuration settings for Hosts within a Management Zone.
- **Tagging** - Key-Value pair for routing tickets, setting ticket priorities, and categorizing hosts

- **Alerting Profile** - Alternative ways of alerting asides from the default of Creating Cherwell Tickets.
- **Custom Log and Metric Events** - Dynatrace has built in infrastructure monitoring. Monitors of Logs and Metrics can be created to alert on.
- **Synthetic** - Dynatrace testing a SSH connection, Web Page, HTTP Request, or other infrastructure for proper functioning

All core, critical, production apps and dependencies are monitored by Synthetics. Test your application, service, or external dependencies via HTTP, Webpage Testing, Linux (SSH), etc.

- Synthetics are tuned to capture meaningful outages of your application. Review your Synthetics or Cherwell Ticket history to check for high false-positive rates.

Dynatrace Problems

1. Failure rate increase

2.

Problem	Impacted	Affected	Root cause	Start date	Duration	profiles
Http monitor global outage P-250710188: Availability	Okta Portals	1		Jul 18 19:32	1 min	Default
Control-M Job Abend: COSDPRP1460 P-250710187: Custom	VMPBMCEMP05.CMUTUAL.com	1		Jul 18 19:29	4 min	ServiceNo...
Pods stuck in pending P-250710145: Resource	aqua-agent	1	aqua-agent	Jul 18 17:31	2 h 3 min	Cherwell, ...
Pods stuck in pending P-250710144: Resource	aqua-agent	1	aqua-agent	Jul 18 17:31	2 h 3 min	Cherwell, ...
Failure rate increase P-250710184: Error	(/CaptainsLogService/V1.0) on ...	1	(/CaptainsLogService/V...)	Jul 18 17:31	2 h 3 min	Consumer, ...
Pods stuck in pending P-250710147: Resource	dynatrace-webhook	1	dynatrace-webhook	Jul 18 17:31	2 h 3 min	Cherwell, ...
Pods stuck in pending P-250710146: Resource	aqua-agent	1	aqua-agent	Jul 18 17:31	2 h 3 min	Cherwell, ...
Pods stuck in pending P-250710143: Resource	dynatrace-webhook	1	dynatrace-webhook	Jul 18 17:31	2 h 3 min	Cherwell, ...
BluePrism Old Session Count P-250710129: Custom	BluePrism SQL Monitor – VMP...	1		Jul 18 16:43	2 h 51 min	Cherwell, ...
Content Services: Composition: PROD ...	VMPDOCGENAP01.CMUTUAL...	1		Jul 18 13:23	6 h 10 min	Cherwell, ...

3.

Dashboard Creation

Step-by-Step Guide to Create a Dashboard in Dynatrace

1. Navigate to Dashboards:

- In the Dynatrace web UI, go to the left-hand menu and click on “Dashboards”.
- Click “Create Dashboard”, give it a name (e.g., *System Resource Monitoring*), and click “Create”.

2. Add CPU Usage Metrics:

- Click “Add tile” → choose “Custom chart”.
- **(Data Explorer)** In the metric selector, search for “CPU usage” (e.g., CPU usage %).
- Choose the aggregation method (e.g., Maximum or Average).
- Optionally, split by host to see per-host CPU usage.
- Click “Add to dashboard”.

3. Add Memory Usage Metrics:

- Repeat the process: Add tile → Custom chart.
- Search for “Memory usage” (e.g., Memory used % or Memory available).
- Configure the chart as needed and add it to the dashboard.

4. Add Disk Usage Metrics:

- Again, go to Add tile → Custom chart.
- Search for “Disk usage” or Disk space used %.
- You can also monitor disk read/write rates if needed.
- Configure and add to the dashboard.

5. Customize Layout:

- Drag and resize tiles to organize the dashboard.
- Use Markdown tiles to add notes or instructions.

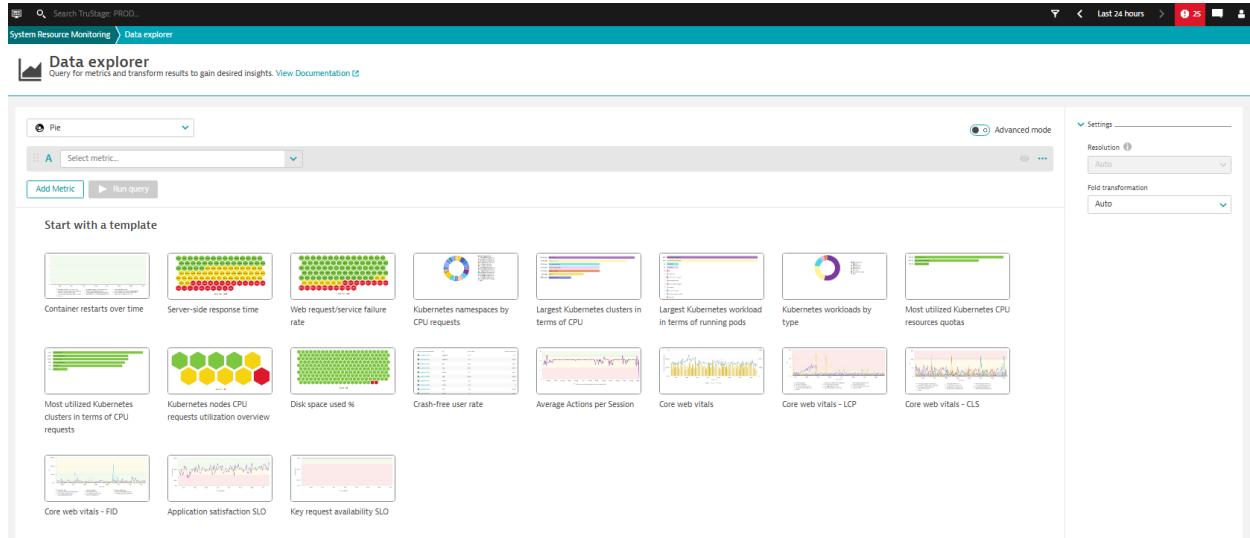
6. Save and Share:

- Click “Done” to exit edit mode.
- Use the Share button to share with teams or set permissions.



Helpful Video Tutorials

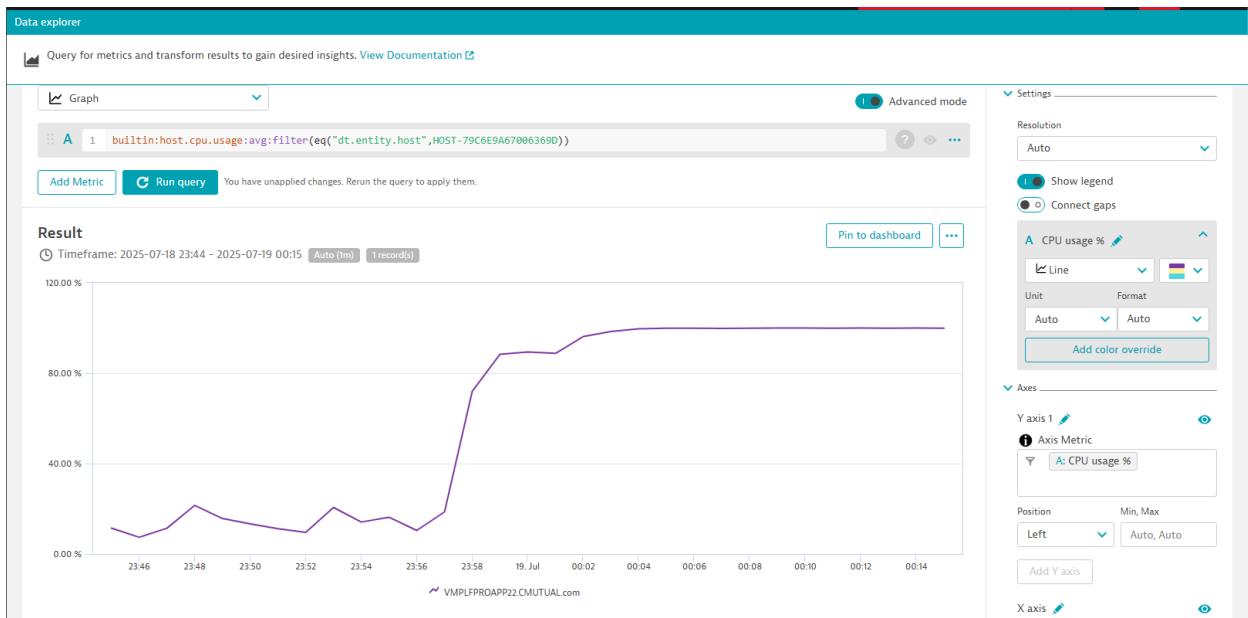
- [Dynatrace Dashboard Creation Tutorial \(YouTube\)](#) – Covers adding CPU, memory, and disk metrics step-by-step.
- [How to Create and Edit Dashboards in Dynatrace](#) – A more comprehensive guide with visualizations and tips.



CPU Usage Metrics

Host	CPU Usage (%)
VAXAISP1PHCMUTUAL.com	87.93 %
VMPLFCFGW002CMUTUAL.com	75.33 %
VMPLFCFGW003CMUTUAL.com	73.48 %
VMPLFCFGW005CMUTUAL.com	68.86 %
VMPLFCFGW007CMUTUAL.com	67.40 %
VMPLFCFGW009CMUTUAL.com	67.00 %
VMPLFCFGW010CMUTUAL.com	65.84 %
VMPLFCFGW011CMUTUAL.com	65.38 %
VMPLFCFGW012CMUTUAL.com	60.16 %
VMPLFCFGW013CMUTUAL.com	58.25 %
VMPLFCFGW014CMUTUAL.com	53.25 %

builtin:host.cpu.usage:avg:filter(eq("dt.entity.host",HOST-79CXXXXXXXXXXXXXX))



CPU Saturation: App Down users unable to login – Live issue.

SQL Client and SQL exception shut down is progress.

saturation high

It shutdown the session and rebooted the server.

Timestamp	Event	Details
2025-07-19 01:40	Process chef-client restarted	▼
2025-07-19 01:37	Process RUNFJ.exe restarted	▼
2025-07-19 01:36	Process SensorLogonTask.exe restarted	▼
2025-07-19 01:36	Host gracefully rebooted: restart – Other (Unplanned)	▼
2025-07-19 01:36	Host shutdown – restart – Other (Unplanned)	▼
2025-07-19 01:02	Process RUNFJ.exe has crashed	▼
2025-07-19 01:02	Process RUNFJ.exe has crashed	▼
2025-07-19 01:00	Process RUNFJ.EXE has crashed	▼
2025-07-19 01:00	Process RUNFJ.EXE has crashed	▼
2025-07-19 00:57	Process RUNFJ.exe has crashed	▼

Show 10 more

Problems

Filter by: Status: Closed | Text: "CPU" | Auto-refresh every 1 min | Pin to dashboard | Clear all

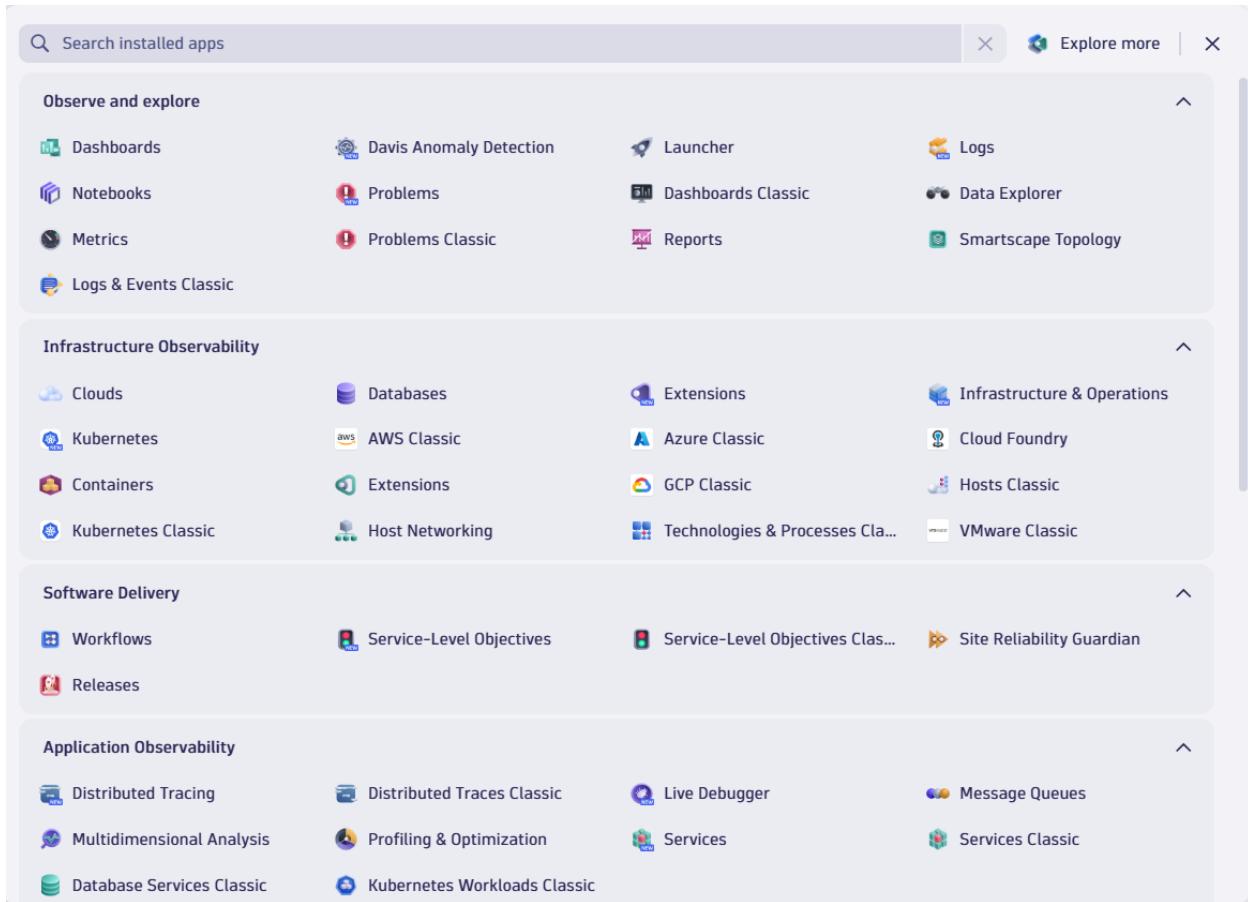
Status: Jul 19 2020 | Open problems: 0 | Closed problems: 6

Severity: Monitoring unavailable, Availability, Error, Slowdown, Resource, Custom | Impact level: Infrastructure, Services, Application, Environment | Maintenance: Any, Under maintenance, Not under maintenance

Filter table (click to interact)

6 Problems

Problem	Impacted	Affected	Root cause	Start date	Duration	Alerting profiles
CPU saturation P-250710283: Resource	VMPITSDBP01.CMUTUAL.com	1	chef-client	Jul 19 00:32	14 min	ServiceNow Ticket...
CPU saturation P-250710277: Resource	VMPMOVEDBP01.CMUTUAL.com	1	VMPMOVEDBP01.CMUTUAL.com	Jul 19 00:01	9 min	
CPU saturation P-250710276: Resource	VMPFLPROAPP22.CMUTUAL.com	1	Windows System	Jul 18 23:59	1 h 37 min	Cherwell, Default, ...
CPU saturation P-250710275: Resource	VMPFLPROAPP21.CMUTUAL.com	1	RUNFJ.EXE	Jul 18 23:36	1 h 32 min	Cherwell, Default, ...
CPU saturation P-250710234: Resource	VMPCDWRAPP05.CMUTUAL.com	1	dialup.exe	Jul 18 21:31	13 min	ServiceNow Ticket...
Azure SQL Database CPU usage % [Azure] P-250710261: Resource	sql-lirs-customer-p01/lirs.Customer_Prod	1	Azure SQL Database (vCore)	Jul 18 20:25	26 min	Cherwell, Default, ...



Failure rate increase

Response time degradation

1 Error event

Dashboard creation use cases

Yes, absolutely — explaining **how you created a Dynatrace dashboard**, and its **use cases** can be a strong point in an interview, especially if you're applying for roles related to DevOps, SRE, cloud monitoring, or system administration.

Here's how you can structure your explanation:

1. Start with the Purpose

Explain **why** you created the dashboard:

“I created a custom dashboard in Dynatrace to monitor system resource usage — specifically CPU, memory, and disk — across our infrastructure. This helped us proactively detect performance bottlenecks and optimize resource allocation.”

2. Describe the Process Briefly

Mention the key steps:

“I used the Dynatrace dashboard editor to add custom chart tiles. For each metric — CPU, memory, and disk — I selected relevant built-in metrics like CPU usage %, Memory used %, and Disk space used %. I configured them to show trends over time and split them by host to identify problematic nodes.”

3. Highlight Use Cases

This is where you show the value:

- **Performance Monitoring:** “We used it to monitor high CPU usage during peak traffic hours.”
 - **Capacity Planning:** “The memory and disk trends helped us plan for scaling and storage upgrades.”
 - **Alerting & Troubleshooting:** “It helped us quickly identify and isolate issues during incidents.”
-

4. Optional: Mention Collaboration or Automation

If applicable:

“I shared the dashboard with the DevOps team and integrated it with alerting policies. I also explored using the Dynatrace API to automate dashboard creation for different environments.”

Script to explain in interviews

Sample Interview Script: Dynatrace Dashboard Creation

“In my previous role, I created a custom dashboard in Dynatrace to monitor key system resources — specifically CPU, memory, and disk usage — across our infrastructure. The goal was to provide real-time visibility into system health and help the team proactively detect performance issues.

I started by creating a new dashboard and added custom chart tiles for each metric. For CPU, I used the CPU usage % metric and configured it to show the average usage over time, split by host. For memory, I tracked both Memory used % and Memory available, and for disk, I included Disk space used % and Disk read/write rates.

One of the key use cases was **performance monitoring** — we could quickly identify which hosts were under heavy load. It also helped with **capacity planning**, as we could see trends in memory and disk usage and plan upgrades accordingly. During incidents, the dashboard was invaluable for **troubleshooting**, helping us isolate the root cause faster.

I also made the dashboard collaborative by sharing it with the DevOps and SRE teams, and I explored using the Dynatrace API to automate dashboard creation for different environments like staging and production.

Overall, it was a great way to centralize observability and improve our operational efficiency.”

Shorter Version

“I created a custom Dynatrace dashboard to monitor CPU, memory, and disk usage across our infrastructure. I used custom chart tiles to visualize key metrics like CPU usage %, Memory used %, and Disk space used %, split by host. This helped us with real-time performance monitoring, capacity planning, and faster incident resolution. I also shared the dashboard with the team and explored automating it using the Dynatrace API.”