Dynatrace Day 3: AI, Database, Synthetic, Cloud & Dashboards – Detailed Notes

# Database Monitoring

**What is Database Monitoring in Dynatrace?**

Database Monitoring ensures that **databases and queries don’t become performance bottlenecks**. Dynatrace automatically detects, monitors, and analyzes **database services, queries, stored procedures, and connections** across your environment.

It answers:

* *Which SQL queries are slowest?*
* *Which services and users are impacted?*
* *Is the issue at the DB level (CPU, locks) or application-level (bad queries, connection pools)?*

**Key Features of Database Monitoring**

**1. Automatic Database Service Detection**

* Dynatrace OneAgent auto-discovers **database instances, clusters, and processes** (MySQL, PostgreSQL, Oracle, SQL Server, DB2, MongoDB, Cassandra, etc.).
* Detects **top queries** without configuration.

**2. SQL & Query-Level Visibility**

* **PurePath® Tracing** captures every database call made by applications.
* Shows **slow queries, high-frequency queries, N+1 query patterns**, and their exact execution time.
* Supports SQL, NoSQL, and cloud-native DBs (Aurora, DynamoDB, Cosmos DB, BigQuery).

**3. Database Health Metrics**

* Monitors:
  + Connections & connection pools
  + Query throughput & response times
  + Index usage & table scans
  + Blocking sessions & deadlocks
  + Lock wait times
  + Cache hit ratios

**4. Infrastructure Monitoring**

* Tracks **host-level metrics**: CPU, memory, disk I/O, network traffic.
* Detects storage latency, slow disks, or VM/container resource issues.

**5. Davis AI Root Cause**

* Correlates **slow user transactions** with **underlying DB issues**.
* Identifies if a regression is due to a query, index issue, or resource saturation.

**6. End-to-End Context**

* Links queries → applications → services → impacted users.
* Example: *“Checkout failed for 2,300 users because the payment service query SELECT … had high lock wait times.”*

**Example Use Cases**

1. **Slow Queries**  
   Dynatrace identifies a specific SELECT \* FROM Orders query taking 5s and highlights missing indexes.
2. **Connection Pool Issues**  
   Detects app-side connection pool exhaustion, traced back to DB spikes.
3. **Blocked Transactions**  
   Flags deadlocks and blocking sessions in SQL Server or Oracle.
4. **Kubernetes & Cloud DBs**  
   Monitors managed DB services (AWS RDS, Azure SQL, GCP Cloud SQL) alongside pods and services.

**Benefits**

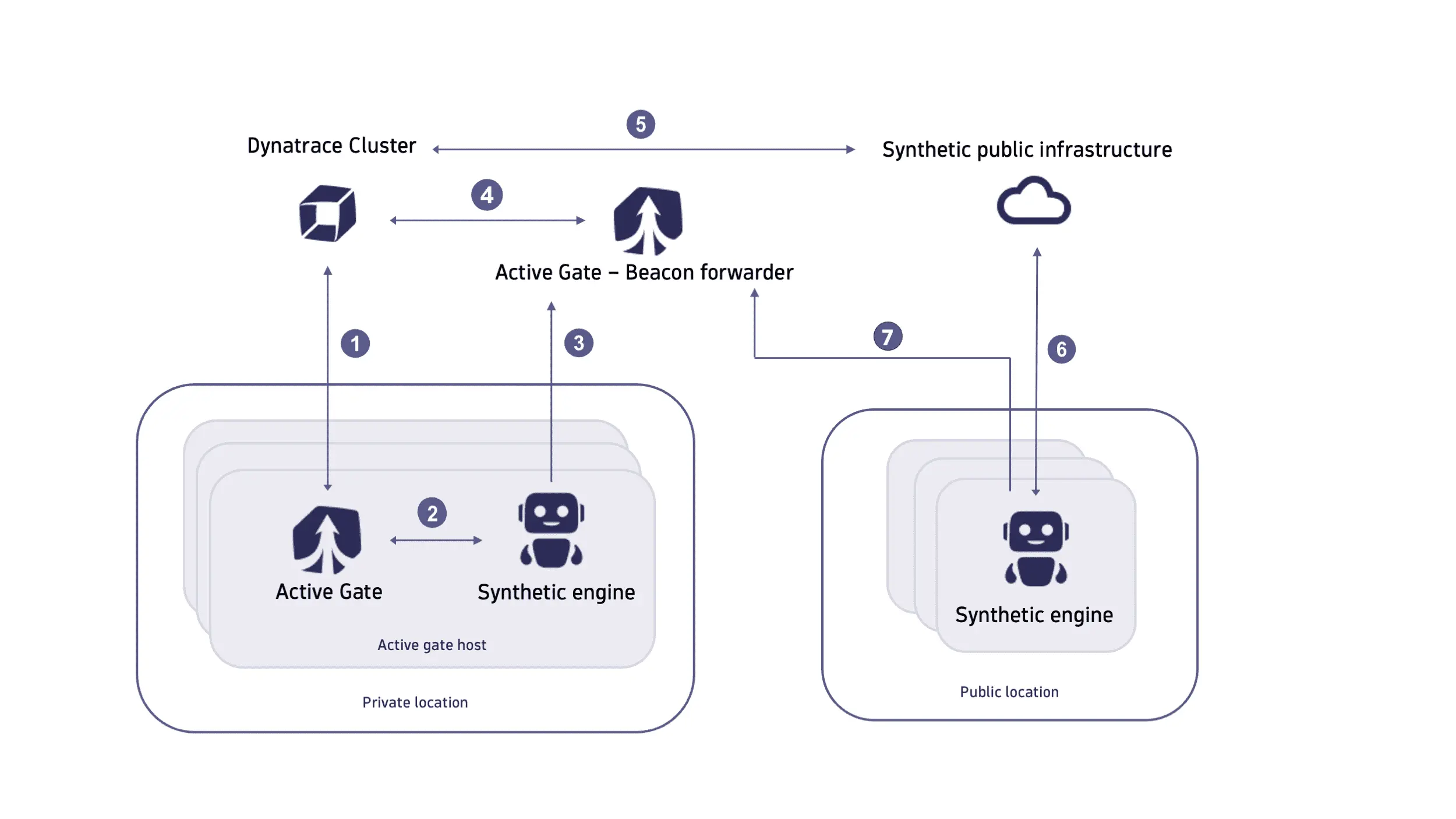
| **Feature** | **Benefit** |
| --- | --- |
| **Query-level tracing** | Pinpoint slow SQLs without manual profiling |
| **Automatic detection** | No config needed—OneAgent discovers DBs |
| **Full-stack correlation** | See DB issues in the context of apps & users |
| **AI-powered RCA** | Faster resolution, less time wasted on false leads |
| **Cloud-native support** | Monitors AWS RDS, Aurora, Cosmos DB, BigQuery, etc. |

# Synthetic Monitoring

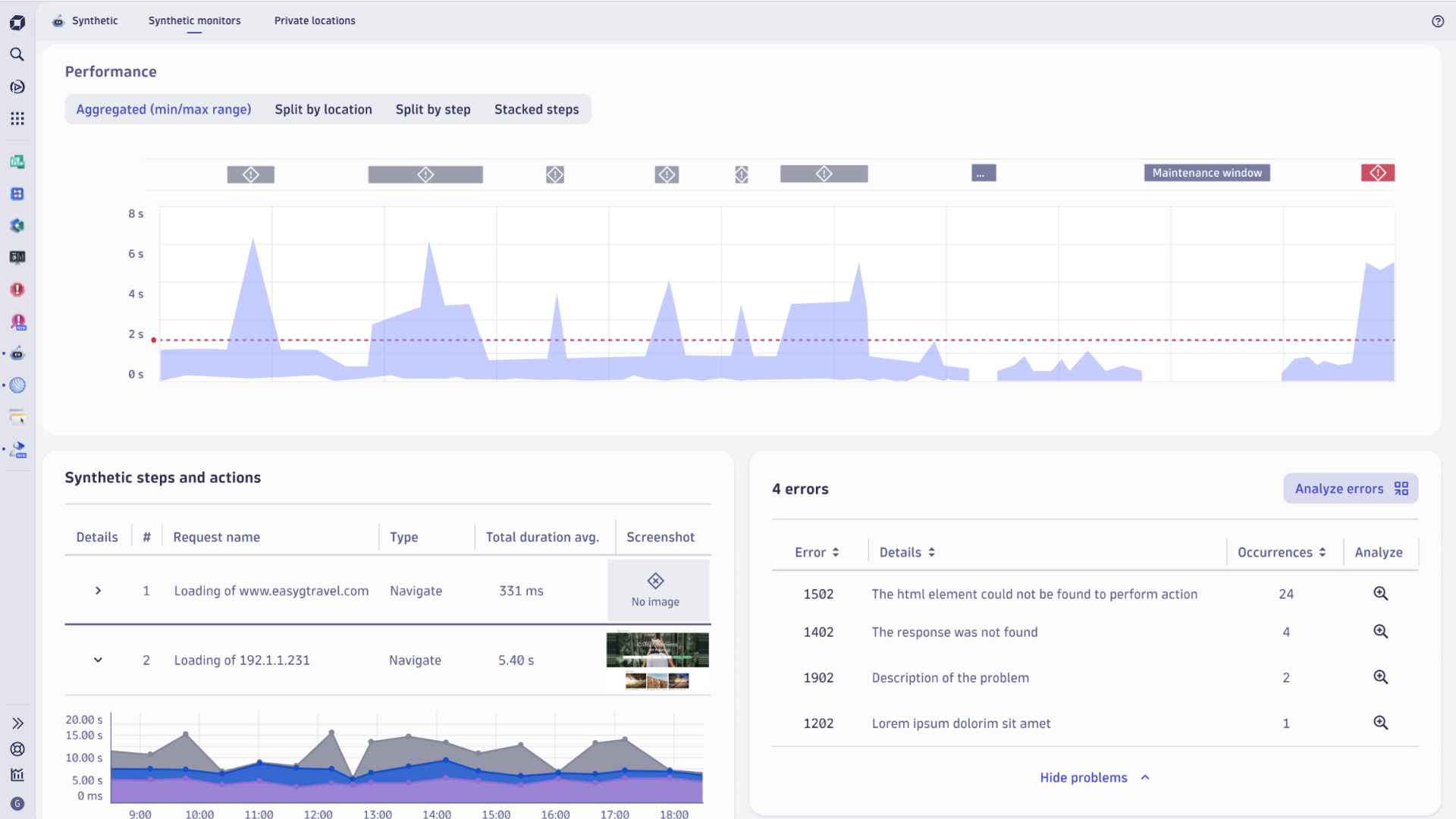
Dynatrace Synthetic Monitoring operates—both in terms of system architecture and practical dashboard views:

**Visual Breakdown**

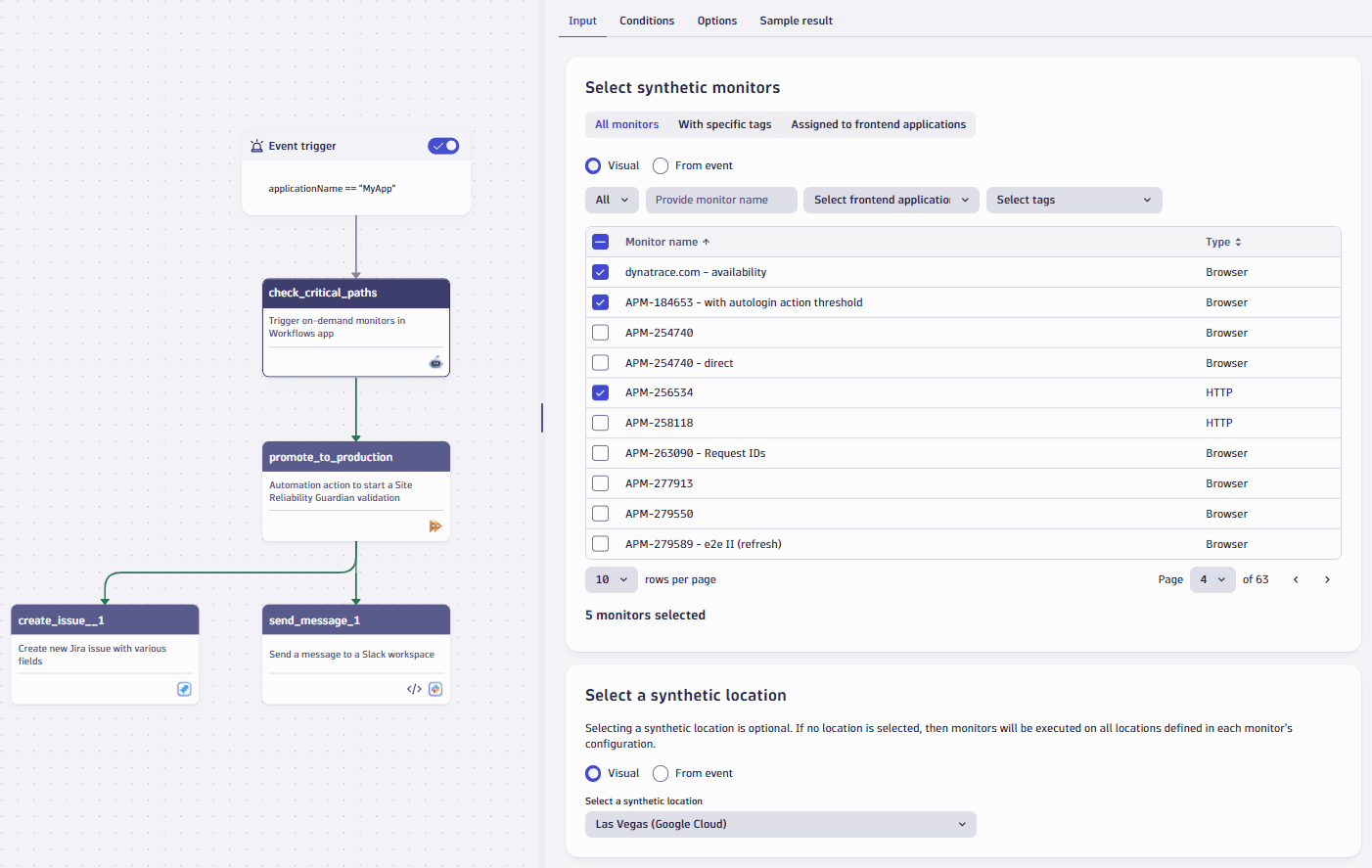
1. **Synthetic Monitoring Architecture (Top‑Left)**  
   This diagram reveals the system’s communication flow between **public and private synthetic locations**, **ActiveGate**, **Synthetic engines**, and the **Dynatrace Cluster**. It highlights how test configurations and results are securely transmitted and processed.



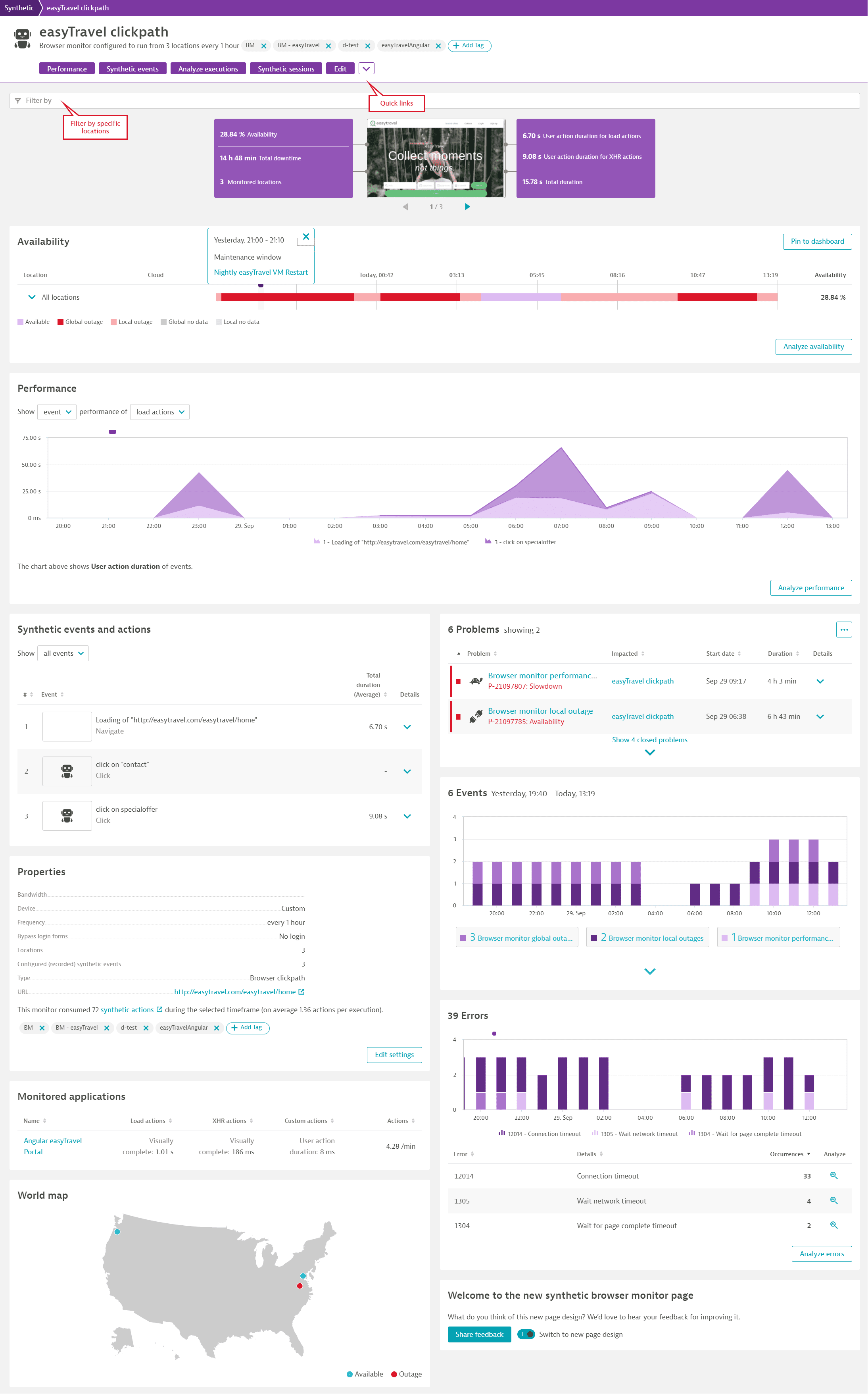
1. **Dashboard Overview (Top‑Right)**  
   A screenshot showcasing the Synthetic Monitoring dashboard. It presents performance and availability trends over time, complete with infographics and quick access to synthetic monitor settings.



1. **Workflows Integration (Bottom‑Left)**  
   Demonstrates Dynatrace’s **Synthetic for Workflows** feature—allowing synthetic tests to be programmatically triggered within automated pipelines. You can choose monitors by tags, applications, or explicit lists and integrate with CI/CD or release workflows.



1. **Synthetic Details View (Bottom‑Right)**  
   Displays the **Synthetic details page** for browser monitors, which includes trend graphs, availability/performance infographics, maintenance indicators, and quick links to filtering, settings, or waterfall analyses.



**How Synthetic Monitoring Works in Dynatrace**

**What It Is**

* Synthetic monitoring—also known as **active monitoring**—emulates user actions to proactively test application workflows and APIs, even before real users interact with them.
* It simulates scenarios like login flows, form submissions, and page navigation under controlled conditions to validate uptime, performance, and functionality.

**Monitor Types**

Dynatrace offers multiple monitor types:

* **Single-URL browser monitors** for availability checks,
* **Browser clickpaths** to validate complex, multi-step user journeys using recorded scripts,
* **HTTP monitors** for API or endpoint testing,
* **NAM (Network Availability Monitoring)** for network-level checks like ping or TCP port availability.

**Features & Capabilities**

* Use a **web-based recorder**—no scripting required—to capture and schedule browser-based synthetic tests from global public locations or private endpoints.
* Supports **CI/CD integration** and **on-demand execution**, enabling “shift-left” testing and release validation before deployment.
* Synthetic monitors generate **problems and alerts** upon performance or availability breaches, with configurable threshold-based notifications.
* By integrating with Dynatrace’s **Workflows**, synthetic tests can power automated validation pipelines—handling tasks like build gating, retries after failure, or notifying teams via Jira/Slack.

**Summary: Why It Matters**

| **Advantage** | **Benefit** |
| --- | --- |
| **Proactive Monitoring** | Detects issues before real users are affected |
| **Global & Internal Coverage** | Synthetic executions from public or private locations |
| **No-Code Test Creation** | Web recorder simplifies test setup without scripting |
| **Pipeline & Workflow Integration** | Enables automated CI/CD validation and recovery processes |
| **Detailed Diagnostics** | Interactive dashboards, screenshots, and waterfall analyses for errors |

# DAVIS AI – Foundation

**What is Davis AI?**

Davis AI is Dynatrace’s **explainable AI (XAI)** engine that continuously analyzes all telemetry—**metrics, logs, traces, user sessions, and events**—to:

* Detect anomalies in real time
* Perform **root-cause analysis** automatically
* Prioritize problems by **business impact**
* Trigger **automated remediation workflows**

Unlike traditional anomaly detection (which often requires threshold tuning), Davis AI works **out-of-the-box** because Dynatrace’s **OneAgent + Smartscape topology** provides complete context of your applications, services, processes, and infrastructure.

**How Davis AI Works**

1. **Data Ingestion**
   * Collects telemetry from apps (Java, .NET, Node.js, etc.), infrastructure (VMs, containers, Kubernetes), logs, and end-user sessions.
   * Builds a **real-time topology model (Smartscape)** of all dependencies.
2. **Anomaly Detection**
   * Uses machine learning & baselining to detect abnormal behavior:
     + Response time spikes
     + Error rate increases
     + Memory/CPU saturation
     + Anomalous user journeys
3. **Root Cause Analysis (RCA)**
   * Instead of just alerting on symptoms, Davis traces the **causal chain** of dependencies.
   * Example:
     + Users see checkout failures → API service errors → caused by DB connection pool exhaustion → caused by a misconfigured deployment.
   * Davis AI **pinpoints the exact root cause**, not just the symptoms.
4. **Business Impact Correlation**
   * Evaluates which user sessions and transactions were affected.
   * Quantifies impact: “1,500 users couldn’t log in” or “Checkout conversion dropped 10%.”
5. **Remediation & Automation**
   * Integrates with **Workflows, ServiceNow, Ansible, Kubernetes, or cloud functions**.
   * Can **auto-heal** incidents (e.g., restart pods, scale services, roll back deployments).

**Example Use Cases**

* **Performance Issue**: Davis detects latency in a microservice, finds it’s due to high GC in a JVM, and flags the root cause as a new deployment.
* **Availability Outage**: Davis spots a failing endpoint, traces it back to a misconfigured DNS entry.
* **Business Impact**: Identifies that a slow payment API is costing a retailer 15% in lost transactions.

**Benefits of Davis AI**

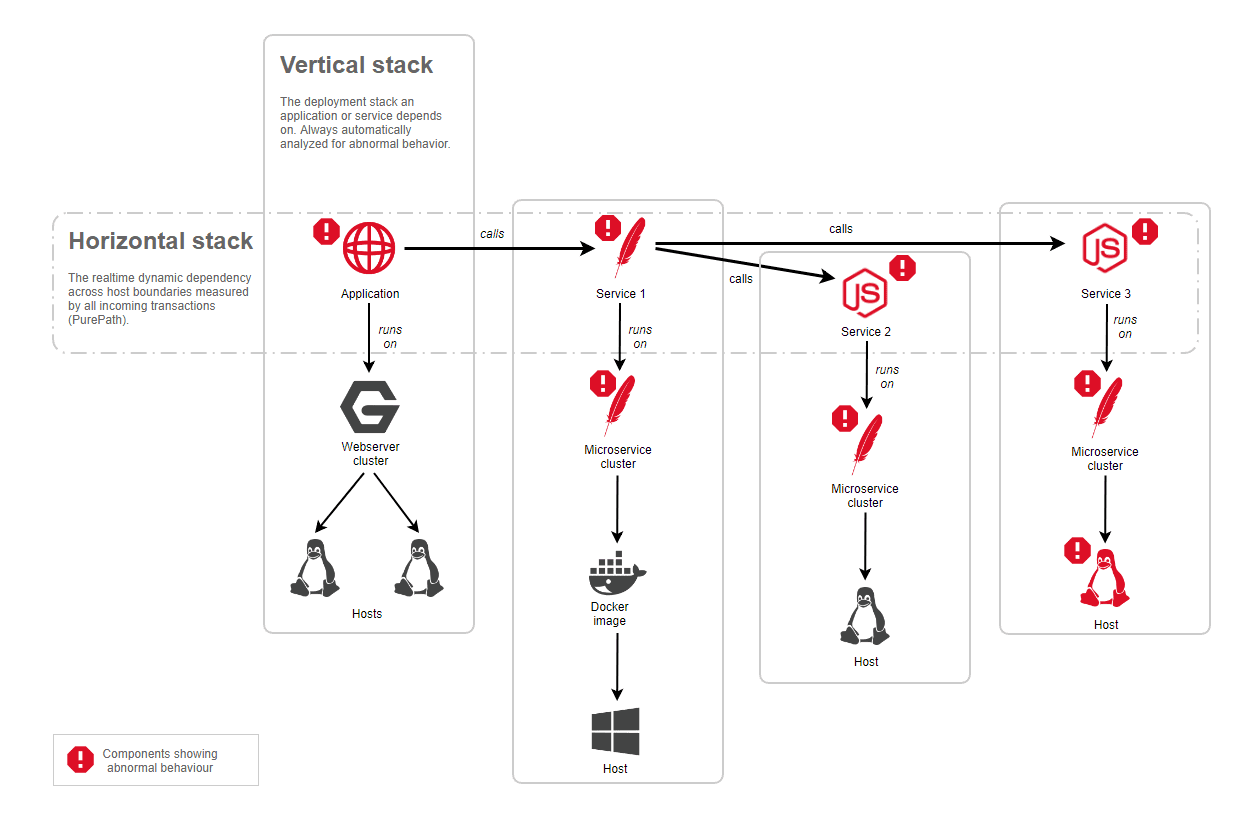
| **Feature** | **Benefit** |
| --- | --- |
| **Automatic RCA** | Cuts down mean-time-to-resolution (MTTR) by pinpointing the cause |
| **Business Context** | Connects IT problems with user and revenue impact |
| **Noise Reduction** | Eliminates alert storms by correlating dependent events |
| **Self-Healing** | Executes automated remediation via workflows or 3rd-party tools |
| **Explainable AI** | Provides transparent reasoning—not a “black box” |

**Visuals You’d Typically See**

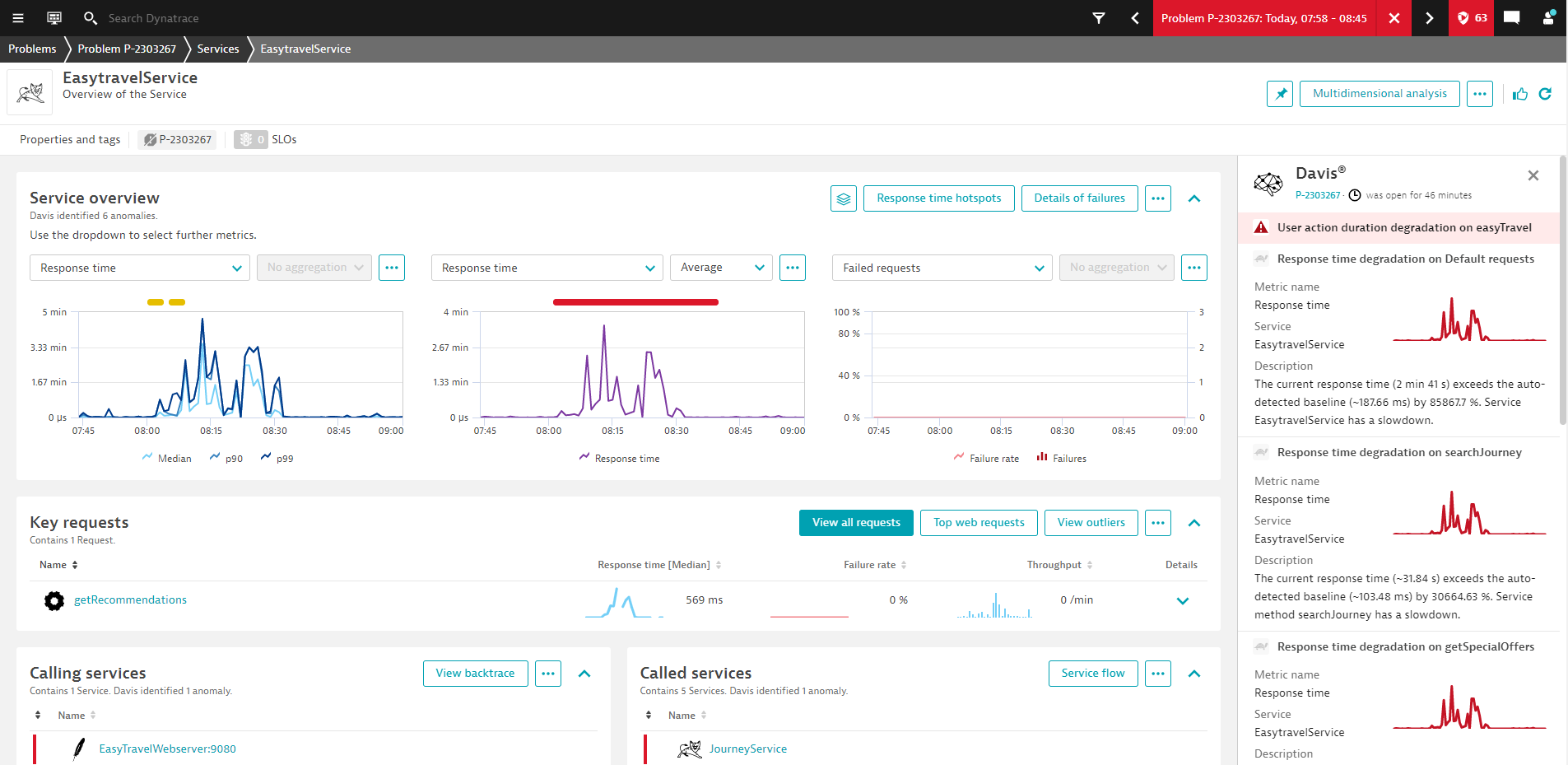
* **Smartscape Topology Map**: Dynamic dependency graph across apps, services, infra.
* **Problem Card**: Davis summarizes the problem, root cause, impacted users, and business impact.
* **Causation Chain Diagram**: Shows the sequence of cause → effect leading to the incident.
* **Workflow Automation Dashboard**: Where Davis recommendations trigger auto-remediation.

Here are four visuals showcasing how **Davis AI** powers root-cause detection and interactive analysis within Dynatrace:

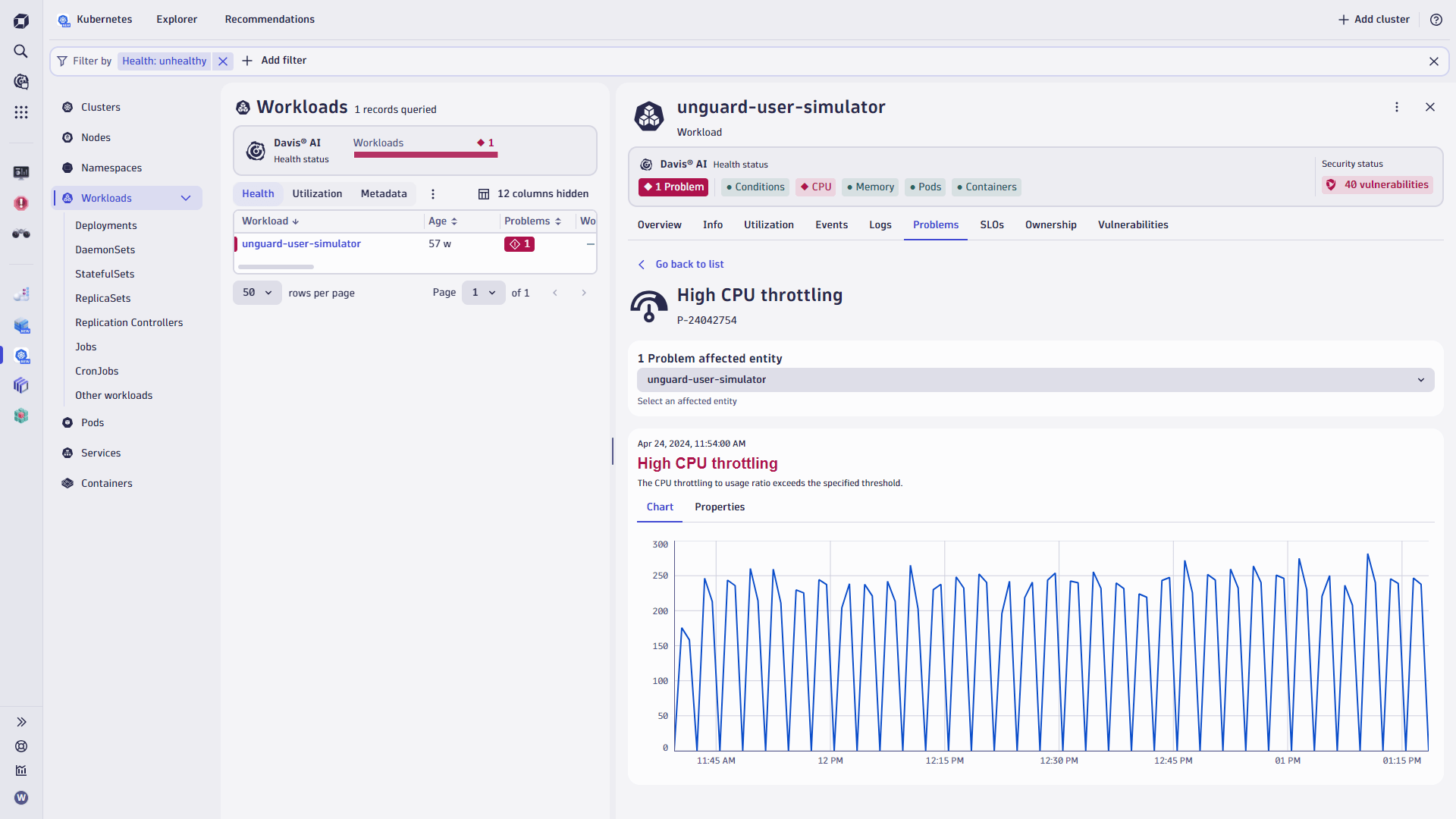
1. **Causal Fault-Tree Diagram**   
   This conceptual flow illustrates how Davis traverses vertical and horizontal dependencies—services, infrastructure, code—to determine the root cause of a problem. It ranks contributors and lets you drill down to code-level insights like failing methods or high garbage collection activity.



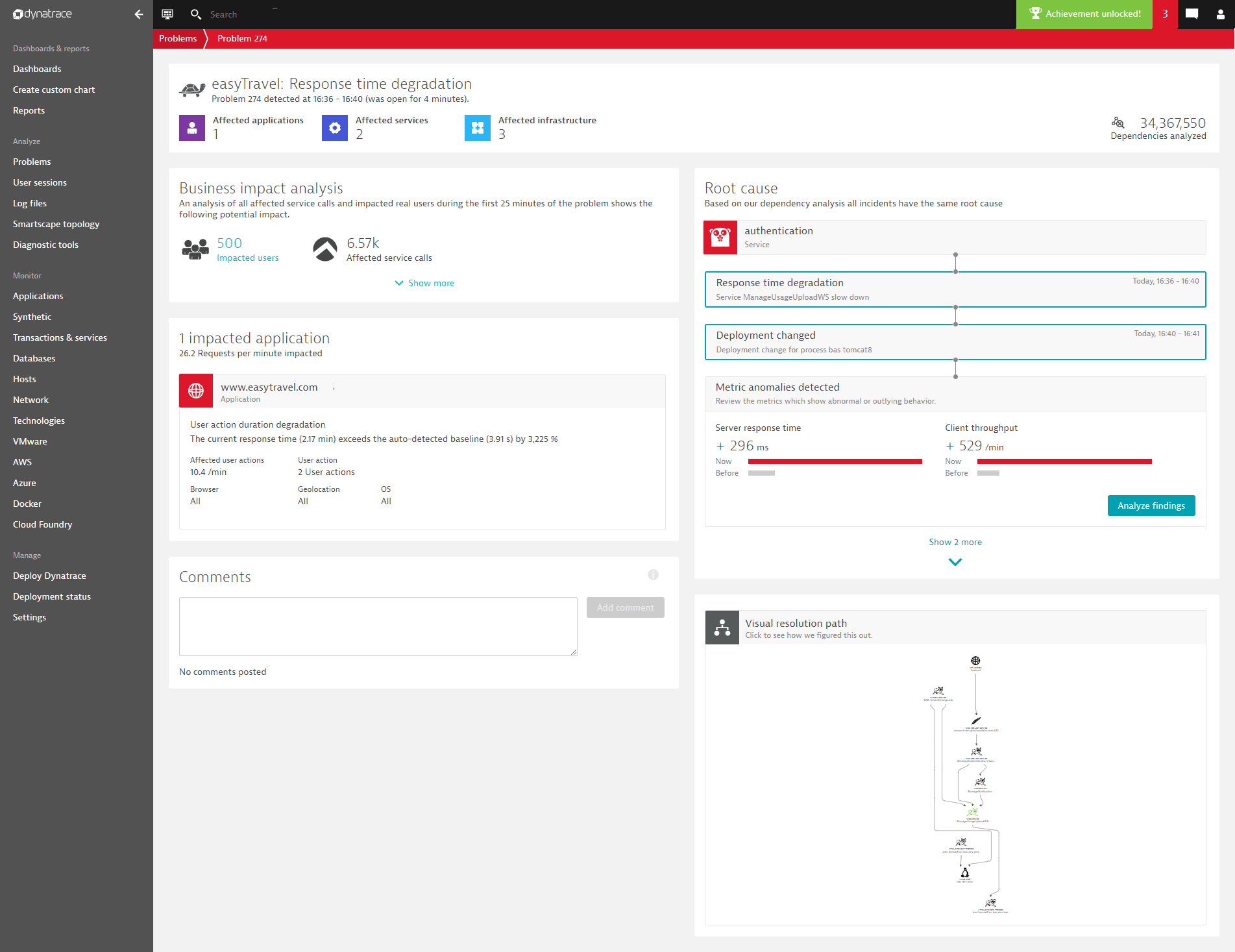
1. **Interactive Davis Problem Mode**  
   Here, Davis overlays visual cues—red for event durations and yellow for metric anomalies—directly onto UI elements. The side panel provides a concise summary of AI findings. This interactive guidance helps you stay focused and context-aware as you navigate through layers of detail.



1. **Context-Aware Root Cause in Infrastructure App**   
   Davis embeds root-cause insights within familiar contexts—like Kubernetes infrastructure or I&O views. You don’t need to switch apps to understand issues like CPU throttling or pod-level failures.



1. **Grouped Root Cause Analysis View**   
   This screenshot reveals how Davis aggregates findings across multiple service instances. Each tile shows metrics with anomalies for clustered services, helping you see patterns or outliers across your stack.



**How Davis AI Enhances Problem Detection & Troubleshooting**

**Causation Over Correlation**

Davis uses Smartscape topology and fault-tree methodology to identify root causes, not just symptomatic alerts. It traces the origin of issues across services and infrastructure to pinpoint the true cause.

**Problem Feed & Real-Time Incident Handling**

* **Unified alerts**: Multiple related Davis events are aggregated into a **single problem**, avoiding alert fatigue.
* **Lifecycle tracking**: Problems are updated in real-time as new events arise, with logic for reopening or marking duplicates.

**Interactive Solving Workflow**

* Davis flags anomalies visually—yellow for change points, red for event duration—across navigational headers and dashboards.
* The **side panel** follows you through the analysis journey, offering summaries, links, and context-aware insights on the go.

**Context-Rich Analysis & Automation**

* **Logs, traces, metrics integrated** for contextual root cause analysis. Davis surfaces relevant logs directly from the problems page to speed diagnostics.
* **Embedded context**: Root-cause insight is available within Infrastructure or Kubernetes views—no need to switch tools.
* Davis supports anomaly detection in Kubernetes—like pod evictions or resource misconfigurations—helping reduce mean-time-to-resolution.

**Expandable & Self-Learning AI Features**

* **Predictive AI**: Forecasts future trends and anomalies.
* **Generative AI (Davis CoPilot)**: Creates notebooks, dashboards, and automations to streamline workflows.

**Summary**

Davis AI in Dynatrace transforms observability with:

* **Precise root-cause analysis** via topology-aware causation detection.
* **Unified problem management**, consolidating alerts into contextual incidents.
* **Interactive, guided workflows** that highlight anomalies and retain context.
* **Rich integration with logs and infrastructure views** for faster troubleshooting.
* **Predictive insights and generative automation**, scaling operations intelligently.

# Dashboards

# Overview: How Dynatrace Dashboards Empower You

# 1. Real-Time, AI-Powered Insights

# Build dashboards instantly using templates and combine metrics, logs, events, traces, and security data in one unified view.

# Leverage Davis AI on dashboards for anomaly detection, forecasting, and reducing alert fatigue with actionable insights—straight from charts.

# 2. Interactive & Rich Visualizations

# Display a variety of tile types:

# Explore tiles (point-and-click to visualize data),

# Query tiles (use Grail/DQL for advanced querying),

# Code tiles (custom data via functions),

# Markdown tiles (for notes, links, formatting, emojis).

# Dashboards support touch interactions, draggable/resizable tiles, mobile-friendly layouts, and rich media like markdown with icons.

# 3. Quick to Build, Easy to Customize

# Begin with ready-made dashboards for common use cases (Infrastructure, Kubernetes, Databases, Synthetic performance, AI workloads) and customize them to fit your needs.[Dynatrace](https://docs.dynatrace.com/docs/analyze-explore-automate/dashboards-and-notebooks/ready-made-documents/ready-made-dashboards?utm_source=chatgpt.com)

# Natural language queries via Davis CoPilot let you add charts without writing DQL. For example: “Show me bytes in and out by host over time, with entity names.”

# 4. Dashboard Management & Sharing

# Organize dashboards using filters like *All*, *Mine*, *Shared with me*, and manage visibility through Management Zones and time frames.

# Dashboards can be shared, cloned, scheduled, and even subscribed to as regular reports (weekly, monthly) for broader distribution.

# Summary Table

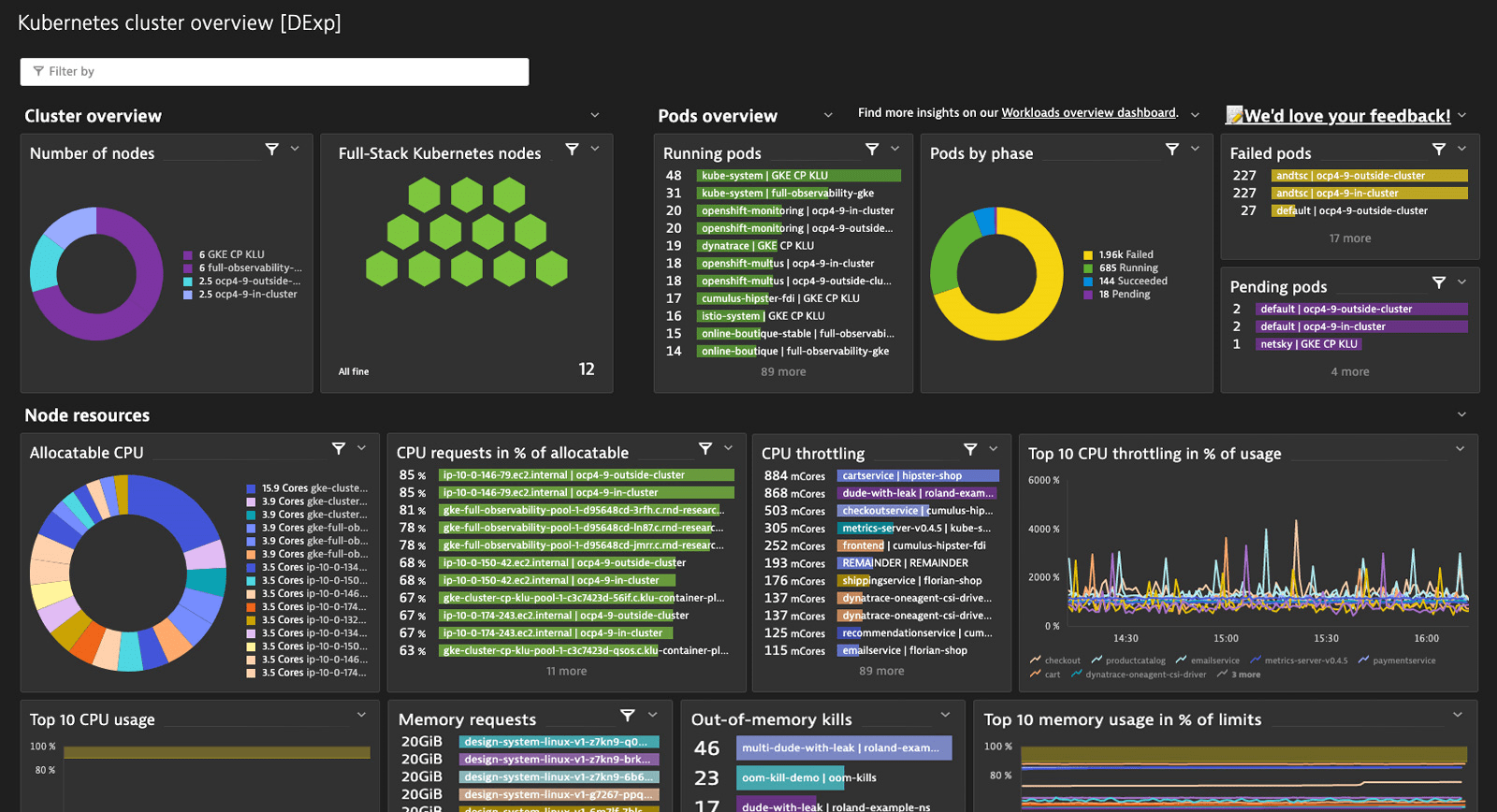
| Feature | Benefit |
| --- | --- |
| AI-Powered Dashboards | Instant insights, anomaly detection, and forecasts using AI |
| Multi-Tile & Visualization Support | Visualize data with explore, query, code, and markdown tiles |
| Ready-Made & CoPilot Support | Fast setup with templates and natural-language chart creation |
| Flexible Sharing & Collaboration | Management zones, templates, scheduling, and report subscriptions |

# Here are four visuals that showcase Dynatrace Dashboards—from an overview to detailed tile layouts and visual customization features:

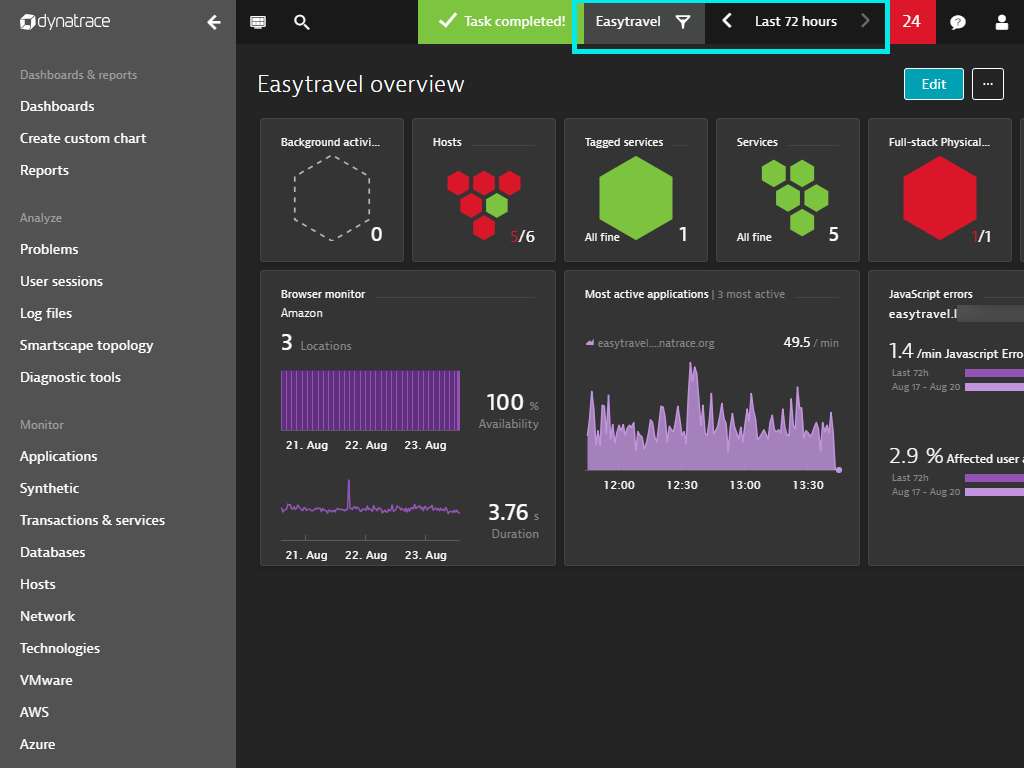
# Top-left: A sleek, AI-powered dashboard highlighting key metrics like service health, performance, and business indicators. It emphasizes Davis AI integration for anomaly detection, forecasting, and alerting.



# Top-right: A richly visual “Data Explorer” dashboard featuring charts like pie, bar, and trend graphs—great for exploring observability metrics and logs effectively.



# Bottom-left: A dashboard with tiles showing health states (green/red hexagons), performance graphs, and filters—demonstrating layout flexibility and user navigation within dashboards.



# Bottom-right: A network overview tile, displaying host and network health visuals alongside traffic metrics—ideal for monitoring network connectivity and issues at a glance.



# Cloud Monitoring

Dynatrace integrates with AWS, Azure, and GCP.  
  
Cloud Monitoring in Dynatrace means continuously observing and analyzing **cloud-native services, workloads, and infrastructure** to ensure performance, availability, and cost-efficiency.

It connects **cloud service APIs**, **infrastructure telemetry**, and **application-level traces** into a **single unified view**, powered by **Smartscape** (real-time topology) and **Davis AI** (root-cause analysis).

**Key Features of Dynatrace Cloud Monitoring**

**1. Automatic Cloud Integration**

* Native support for **AWS, Azure, GCP, Kubernetes, OpenShift, VMware, Oracle Cloud, and more**.
* Auto-discovers cloud resources (VMs, services, databases, storage, messaging, serverless).
* Pulls in **metrics via APIs** + enriches them with **OneAgent telemetry**.

**2. Kubernetes & Container Observability**

* Monitors **pods, nodes, namespaces, workloads, and clusters** automatically.
* Detects restarts, failures, scaling events.
* Provides **deep pod/service visibility** (CPU, memory, restarts, network).
* Links cloud-native workloads to **end-user experience and app traces**.

**3. Service Topology (Smartscape)**

* Visual dependency map of:
  + Applications
  + Services
  + Processes
  + Hosts
  + Cloud resources
* Enables **contextual root-cause analysis**.

**4. Full-Stack Monitoring**

* Covers all layers:
  + **Cloud infrastructure** (VMs, clusters, storage, networking).
  + **Platform services** (databases, load balancers, messaging queues).
  + **Applications** (Java, .NET, Node.js, Python, etc.).
  + **User experience** (RUM + Synthetic monitoring).

**5. Davis AI for Cloud**

* Detects anomalies (e.g., spike in API Gateway errors).
* Links **infrastructure failures → application errors → user impact**.
* Provides **root cause analysis** and **auto-remediation workflows**.

**6. Cost & Resource Optimization**

* Monitors **cloud spend** (via DPS metrics).
* Detects over-provisioning (idle VMs, underutilized nodes).
* Helps optimize **Kubernetes scaling** and **cloud billing efficiency**.

**Example Cloud Use Cases**

1. **AWS Monitoring**
   * Track EC2, RDS, S3, Lambda, EKS.
   * Detect spikes in Lambda cold starts or S3 latency.
   * Correlate with application performance.
2. **Azure Monitoring**
   * Monitor App Services, AKS, Functions, SQL Database.
   * Alert on resource scaling failures.
   * Analyze API Management performance.
3. **GCP Monitoring**
   * Track GCE, GKE, Cloud SQL, Pub/Sub, BigQuery.
   * Detect anomalies in Dataflow pipelines.
   * Correlate Kubernetes pod restarts with user-facing errors.

**Benefits of Dynatrace Cloud Monitoring**

| **Feature** | **Benefit** |
| --- | --- |
| **Unified Observability** | One view for multi-cloud + hybrid |
| **Automatic Discovery** | No manual setup — OneAgent + API integrations |
| **Smartscape Topology** | Contextual root-cause tracing across services |
| **Davis AI** | Cuts MTTR by auto-detecting true causes |
| **Cost Awareness** | Tracks resource usage vs. spend |
| **DevOps Integration** | CI/CD & auto-remediation support |

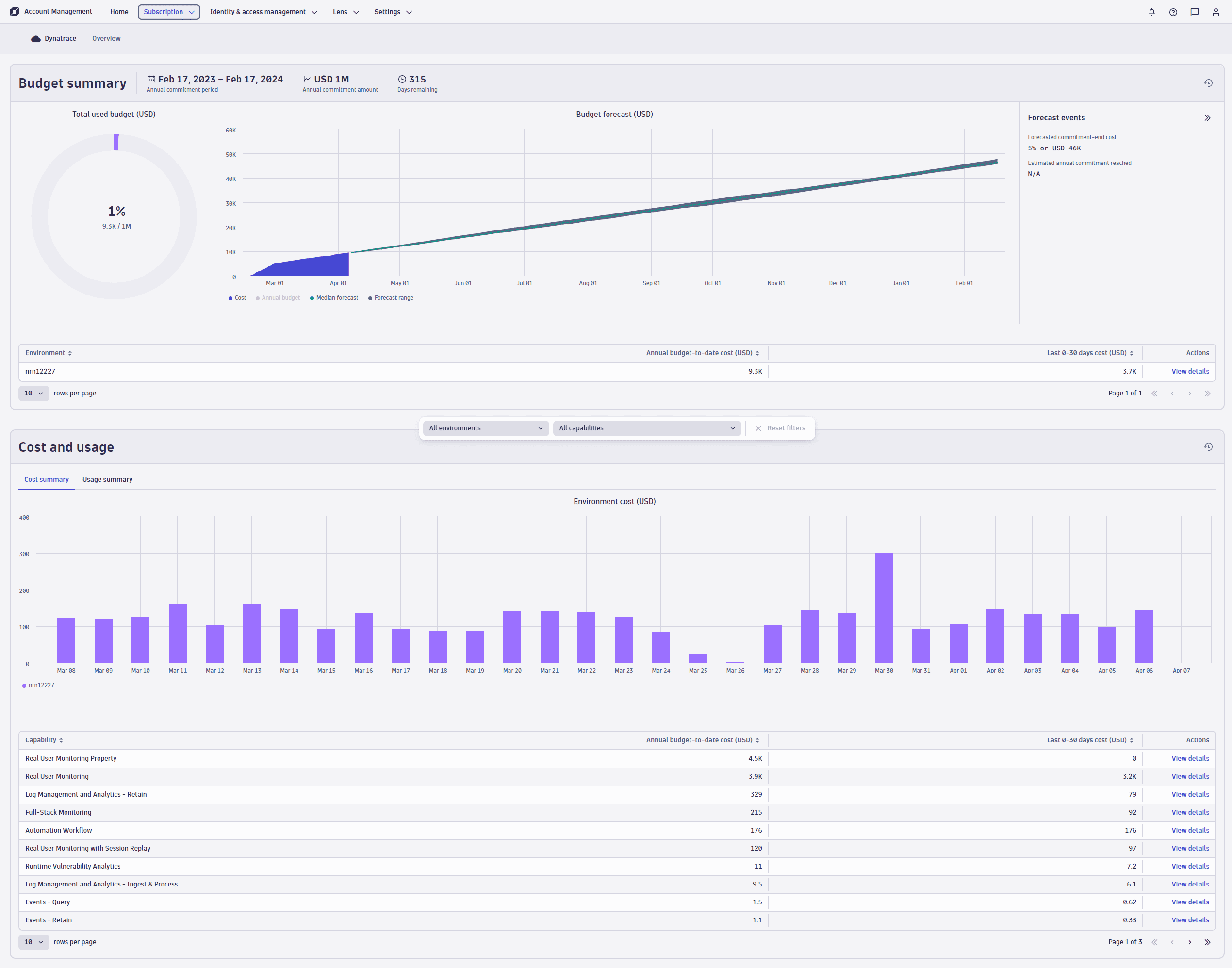
# Licensing & Consumption

Understanding usage is key to cost control in Dynatrace.  
  
**Dynatrace Licensing & Consumption: Diagram & Screenshot Overview**

**1. Dynatrace Platform Subscription (DPS) Model – Visual Snapshot**

The image displays:

* A **donut chart** showing how much of your annual subscription budget has already been used, with a breakdown by monitored environments.
* A **line chart** that illustrates the trajectory of costs incurred over time versus the yearly commitment, also forecasting future spend.
* A **detailed environment table** listing usage and cost data per environment, including current usage, last 30-day change, and forecasted consumption.



**Licensing & Consumption: How It Works**

**A. Dynatrace Platform Subscription (DPS) – Flexible, Usage-Based Licensing**

* **Capability-based pricing**: DPS bills based on actual consumption of platform features (e.g., Full-stack Monitoring, RUM, Synthetic) rather than predefined bundles.
* Organisations agree to a minimum **annual commitment**, usually for 1–3 years.
* Real-time tracking of consumption is available via:
  + **Account Management UI** (Subscription → Accounting)
  + **DPS API**
* Once the **annual commitment is reached**, continued usage incurs **on-demand billing** at the same published rates.

**B. Budget, Forecasting & Breakdown Tools**

* **Budget Summary View** (as shown in image):
  + Tracks total cost vs. annual commitment.
  + Shows cost forecasting—if projected usage exceeds the budget, the expected breach date is flagged.
  + Breaks down usage by environment for granular insight.
* **Detailed Cost & Usage Breakdown** (last 30 days):
  + Offers drill-downs by capability and environment.
  + Includes charts and tables for deeper inspection of trends and anomalies.
  + You can filter between cost views and usage views per capability.
* **Capability-Level Analysis**:
  + Allows inspection of cost vs. usage for a specific capability, displayed across various environments and timeframes.

**C. Classic Licensing (Legacy) Terms**

* Prior to DPS, licensing was structured around discrete **units**:
  + **Host Units**: Billing based on monitored hosts (and their RAM).
  + **DDUs (Davis Data Units)**: Used for custom metrics, logs, traces.
  + **DEM Units**: Applied to Synthetic Monitoring, RUM, Session Replay.
  + **ASUs (Application Security Units)**: For Application Security features.

**At a Glance: Licensing Models Compared**

| **Licensing Model** | **Billing Mechanism** | **Key Feature** |
| --- | --- | --- |
| **Dynatrace Platform Subscription (DPS)** | Usage-per-feature with annual commitment, auto forecasting | Flexible, scalable to usage growth, transparent |
| **Legacy Classic Licensing** | Predefined units such as Host Units, DDUs, etc. | Simpler but rigid and less adaptive |

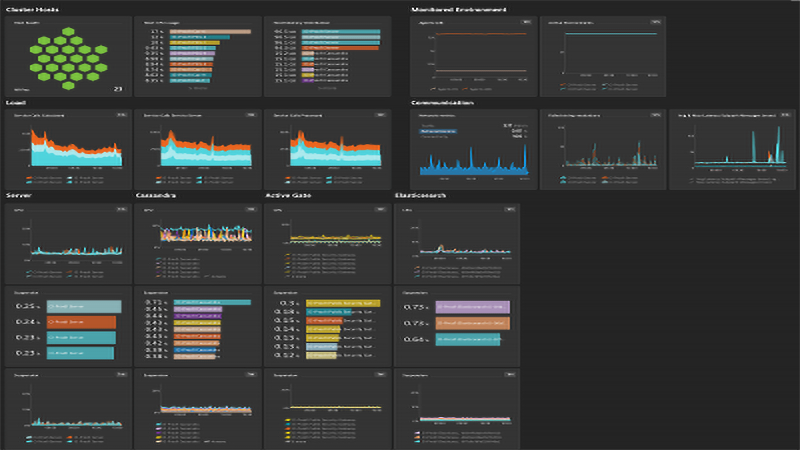
**Ready to Explore?**

You can explore the DPS views yourself right from the Dynatrace web UI:

1. Navigate to **Subscription → Accounting** in **Account Management**.
2. Review the **Budget Summary**, **Cost & Usage Breakdown**, and deep-dive into **Capability / Environment Analysis**.
3. Use these insights to identify cost spikes, optimize usage—e.g., reduce unnecessary hosts, switch monitoring modes, or tune trace volumes.

# Deployment Status Monitoring Deployment Status Monitoring in Dynatrace provides continuous visibility into application deployments by combining:

* **Version detection** → Dynatrace automatically detects and tags new deployments (via OneAgent auto-instrumentation or metadata from CI/CD).
* **Metrics & Traces** → It monitors performance, resource usage, and errors per deployment version.
* **Smartscape & Davis AI** → Automatically correlates changes to problems and identifies whether a deployment is the root cause.
* **Release events integration** → CI/CD tools (Jenkins, GitLab, Azure DevOps, Argo, etc.) can push deployment events to Dynatrace for richer context.



**Key Features**

1. **Automatic Deployment Detection**
   * OneAgent identifies when a new service or process version is deployed.
   * Version info is captured from metadata (JARs, DLLs, containers, K8s labels, etc.).
2. **Release Event Integration**
   * Dynatrace APIs accept release metadata from CI/CD systems.
   * Events are shown in the service timeline for full deployment traceability.
3. **Deployment Status Dashboard**
   * Dedicated view showing active, successful, or failing deployments.
   * Provides error rate, latency, and availability comparison between old vs. new versions.
4. **Davis AI Correlation**
   * Detects anomalies (e.g., spike in errors after release).
   * Identifies if the root cause is tied to the new deployment.
   * Helps accelerate rollback decisions.
5. **Business Impact View**
   * Links deployment issues with **real-user monitoring (RUM)** sessions.
   * Quantifies impact: “1,200 users affected after version 3.2 rollout.”

**Example Use Cases**

* **Monitor a canary deployment** → Compare metrics between old and new versions before full rollout.
* **Detect regression after release** → Dynatrace alerts if latency or failure rate increases post-deployment.
* **Kubernetes workload monitoring** → Automatically shows pod restarts, scaling events, and unhealthy rollouts.
* **Automated rollback trigger** → CI/CD pipelines can use Dynatrace problem events to auto-revert failed deployments.

**Benefits**

| **Feature** | **Benefit** |
| --- | --- |
| Automatic version awareness | Zero manual tagging needed |
| Real-time error/performance tracking | Detect regressions instantly |
| Davis AI root-cause analysis | Pinpoints if deployment caused an incident |
| CI/CD integration | Seamlessly fits into release pipelines |
| Business correlation | Measures user & revenue impact |