AppDynamics

Application Performance Monitoring for US Government

AppDynamics GovAPM User Guide

APPDYNAMICS

AppDynamics is now part of Cisco.

CISCO

DOCUMENT REVISION HISTORY

Date	Description	Version	Author
8/30/2019	Initial Draft	1.0	AppDynamics

TABLE OF CONTENTS

Contents

1 INTRODUCTION

This User Guide provides the information necessary for Government users to effectively use the AppDynamics Application Performance Monitoring for US Government system (AppDynamics GovAPM).

The AppDynamics GovAPM User Guide describes the AppDynamics' GovAPM service. Reading this guide will provide you with an overview of the product, instructions for setup and configuration, instructions for using the GovAPM user interface (UI) and troubleshooting information to solve common issues.

2 OVERVIEW OF APPDYNAMICS GOVAPM

AppDynamics GovAPM is an application performance monitoring (APM) solution. AppDynamics GovAPM provides end-to-end visibility into the performance of your applications.

AppDynamics GovAPM enables users to:

- Troubleshoot problems such as slow response times and application errors
- Automatically discover application topology and how components in the application environment work together to fulfill key business transactions for its users
- Measure end-to-end business transaction performance, along with the health of individual application and infrastructure nodes
- Receive alerts based on custom or built-in health rules, including rules against dynamic performance baselines that alert users to issues in the context of business transactions
- Analyze applications at the code execution level using snapshots

AppDynamics GovAPM comprises of three (3) key components:

- **Controller:** The Controller service is the primary interface users use to monitor, troubleshoot, and analyze your entire application landscape, from backend infrastructure and servers to the end users' app.
- **End User Monitoring (EUM)**: The EUM service provides end-to-end visibility on the performance of web browser applications. The EUM service helps troubleshoot problems such as slow web or Ajax. EUM provides metrics on application performance and user activities.
- **Events Service:** In the AppDynamics model, events represent a change in the state of a monitored application. An event can represent an error or exception generated by the application, the crossing of a performance threshold, or an operational change in the application. Users can define policies that generate notifications or perform actions when an event occurs. The Events Service serves as a backend for data collection and allows for upload of custom data sources (logs, etc.).

GovAPM Deployment

AppDynamics GovAPM is APM delivered as software as a service (SaaS). In this deployment model (as illustrated in Figure 2-1), all AppDynamics GovAPM services run as SaaS and agents are configured to talk to the GovAPM SaaS endpoints.

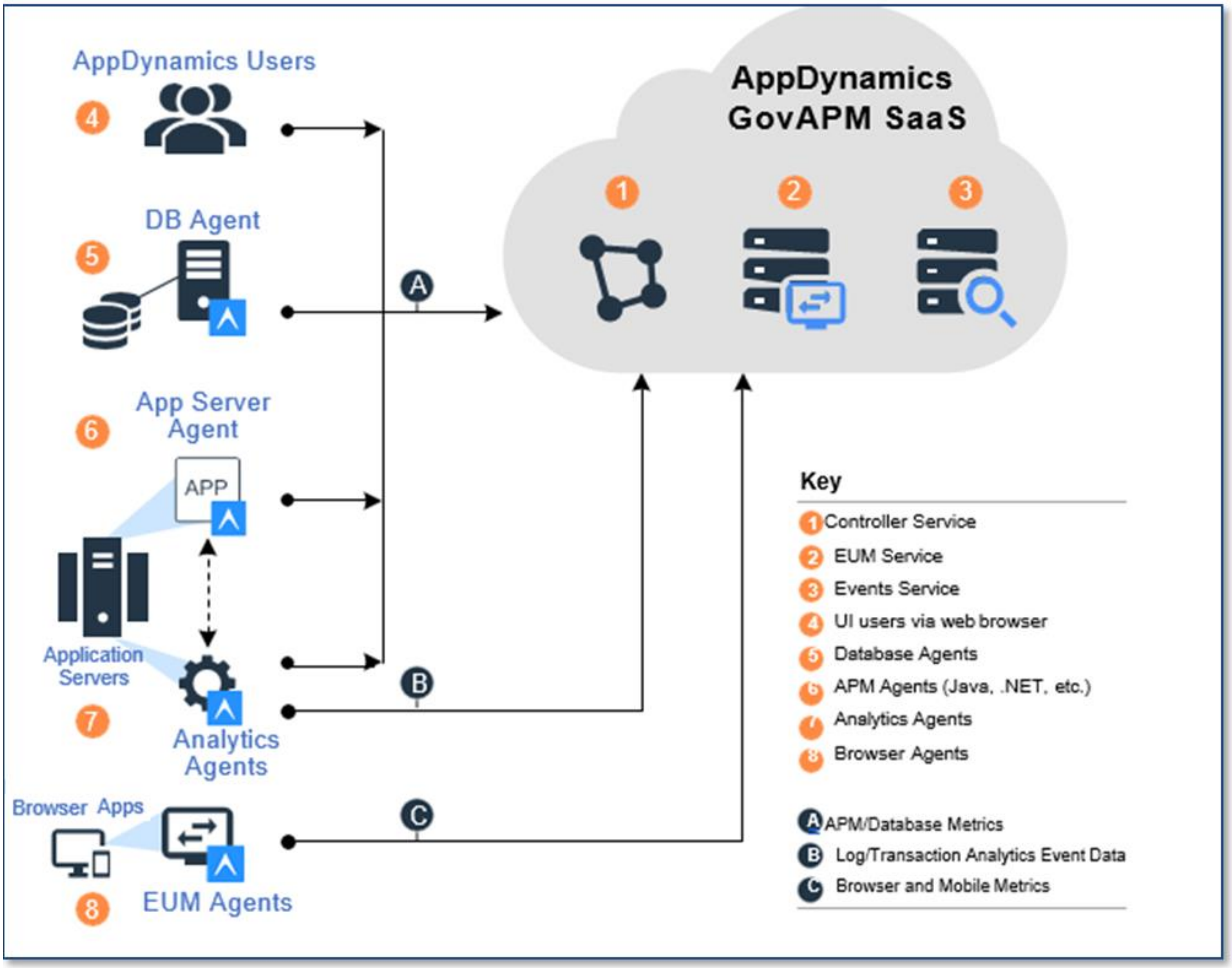


Figure 2-1. AppDynamics GovAPM SaaS Deployment

Application Model Overview

The first step in using AppDynamics GovAPM for monitoring your application environment is to get acquainted with how AppDynamics models an application environment. The model of your application environment will serve as the framework around which AppDynamics organizes and presents performance information.

A typical application environment consists of the different components that interact in a variety of ways to fulfill requests from the application's users:

- Web applications served from an application server
- Databases or other data stores
- Remote services such as message queues and caches

AppDynamics app agents automatically discover the most common application frameworks and services. Using built-in application detection and configuration settings, agents collect application data and metrics to build flow maps.

A flow map visually represents the components of your application to help you understand how data flows among the application components. For example, Figure 2-2 illustrates a business transaction flow map for a simple e-commerce application, showing data flowing between web services, message queues, and databases.

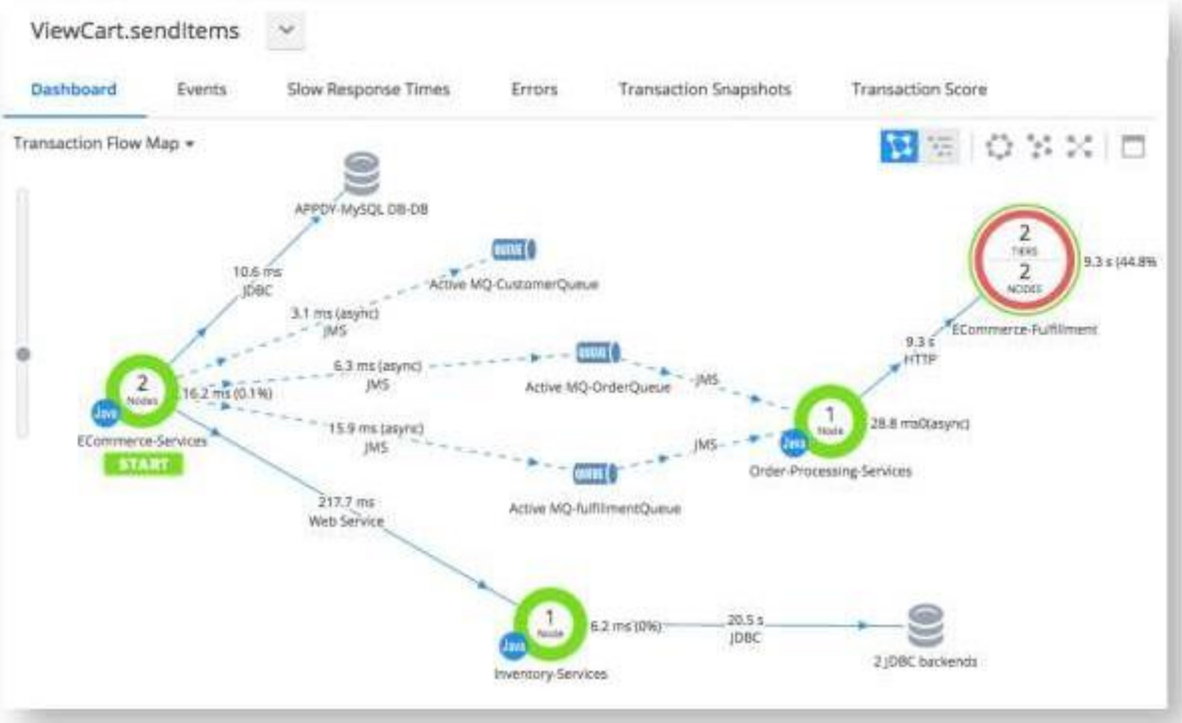


Figure 2-2. A Sample Flow Map

Business Transactions

In the AppDynamics model, a business transaction represents the data processing flow for a request, most often a user request. In real-world terms, many different components in your application may interact to provide services to fulfill the following types of requests:

- In an e-commerce application, a user logging in, searching for items or adding items to the cart
- In a content portal, a user requests content such as sports, business or entertainment news

AppDynamics app agents discover requests to your application as entry points to a business transaction. Similar requests, such as user login, are treated as multiple instances of the same business transaction. The agents tag the request data and trace the request path as it passes from web servers to databases and other infrastructure components. AppDynamics collects performance metrics for each tier that processes the business transaction.

Because AppDynamics orients performance monitoring around business transactions, you can focus on the performance of your application components from the user perspective. You can quickly identify whether a component is readily available or if it is having performance issues. For instance, you can check whether users are able to log in, check out or, view their data. You can see response times for users, and the causes of problems when they occur.

Business Applications

A business application is the top-level container in the AppDynamics model. A business application contains a set of related services and business transactions.

The best way to organize business applications for you depends on your environment. A leading consideration for most cases, however, is to organize business applications in a way that reflects work teams in your organization, since role-based access controls in the Controller UI are oriented by business application.

Nodes

A node in the AppDynamics model corresponds to a monitored server or JVM in the application environment. A node is the smallest unit of the modeled environment. Depending on the agent type, a node may correspond to an individual application server, JVM, CLR, PHP application, Apache Web server.

Each node identifies itself in the AppDynamics model. When you configure the agent, you specify the name of the node, tier, and business application under which the agent reports data to the Controller.

Tiers

A tier is a unit in the AppDynamics model composed of a grouping of one or more nodes. How you organize tiers depends on the conceptual model of your environment.

Often a tier is used to a group of a set of identical, redundant servers. But that is not strictly required. You can group any set of nodes, identical or not, for which you want performance metrics to be treated as a unit into a single tier.

The single restriction is that all nodes in a single tier must be the same type. That is, a tier cannot have mixed types of agents, such as both .NET and Java nodes.

The traffic in a business application flows between tiers, as indicated by lines on the flow map, which are annotated with performance metrics.

In the AppDynamics model:

- There is no interaction among nodes within a single tier
- An application agent node cannot belong to more than one tier

Backends

A backend is a component that is not instrumented by an AppDynamics agent but that participates in the processing of a business transaction instance. A backend may be a web server, database, message queue, or another type of service.

The agent recognizes calls to these services from instrumented code (called exit calls). Performance information is available for the backend call.

End User Monitoring (EUM)

With End-User Monitoring (EUM) component integrated with Application Monitoring, you can correlate business transaction performance to the user experience for those transactions.

If server app agents run on the applications that serve your browser applications, you can further configure the app server agents to inject JavaScript agent into the code that runs on the browser.

Application Monitoring and Analytics

For those times when tracing application code does not provide enough clues to track down the cause of a problem, AppDynamics provides visibility into the transaction logs that can be correlated to specific business transaction requests.

3 GETTING STARTED

User Access Considerations

All AppDynamics GovAPM customers are government entities and their authorized contractors; and, are only granted access to the services through a formal contracting process. The contracting process establishes the relationship between AppDynamics and the customer, defines the customer’s intended use of the service, authorizes the use of AppDynamics GovAPM and ensures the proper allocation of resources.

Accessing the GovAPM SaaS

Log in to the Controller User Interface (UI): The GovAPM Controller UI is the primary interface for configuring and monitoring your application environment.

For accessing the GovAPM Controller UI, you will be contacted by our provisioning team to get set up. You will need to show (via a WebEx) photo identification before you receive your login information.

The AppDynamics GovAPM UI is an HTML 5-based browser application that works best with the latest version of any modern browser. The Controller UI has been tested with the following browsers and versions:

- IE 9+
- Safari 6+
- Chrome 16+
- Firefox 6+
- Microsoft Edge

Opera and older versions of Firefox, IE, and Safari browsers may still operate, but some features may not display as intended.

The Controller UI requires Flash Player 10 or greater; AppDynamics recommends version 11.

Certain types of ad blockers can interfere with features in the Controller UI. We recommend disabling ad blockers while using the Controller UI.

Once you get your provisioning information you can login. After clicking "Login", you'll be redirected to AppDynamics.com to enter your AppDynamics GovAPM credentials. If you're already logged into AppDynamics.com, you'll automatically be redirected to the Getting Started Wizard.

Click on "Agent Download Wizard" for instructions on how to deploy AppDynamics agents.

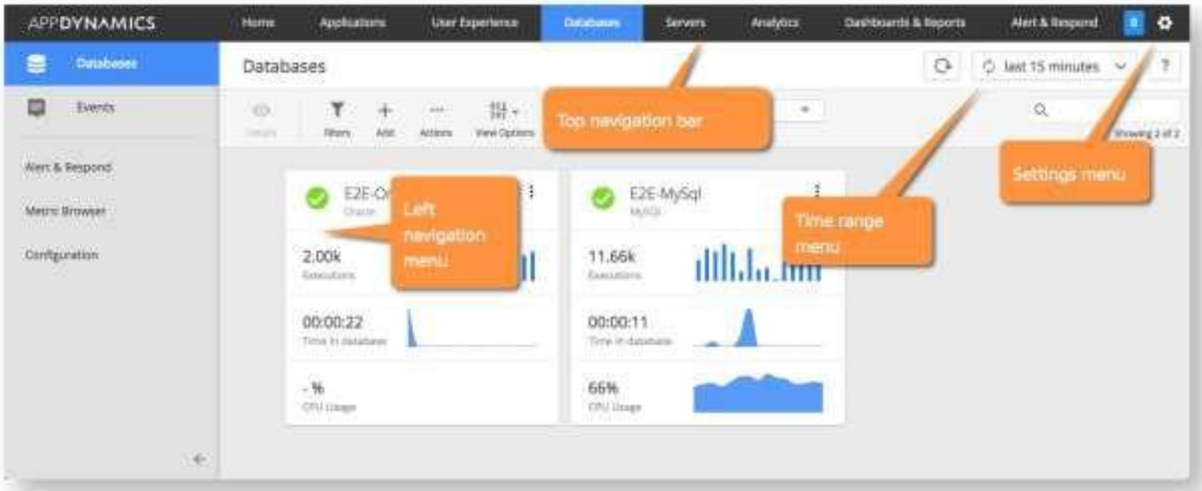
Controller UI Navigation

The GovAPM Controller UI is the primary interface you will use to monitor, troubleshoot, and analyze your entire application landscape, from backend infrastructure and servers to the end user's app.

As you start using the Controller UI to configure AppDynamics and monitor your environment, it is helpful to get acquainted with the elements of the Controller UI that you will use most and that are referenced frequently in this documentation.

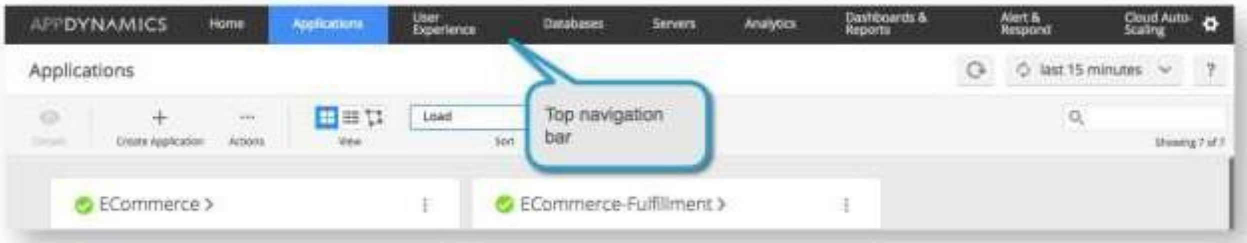
Note that the exact controls and pages that appear for you in the UI depend upon your user permissions.

The following image shows the main elements of the interface:



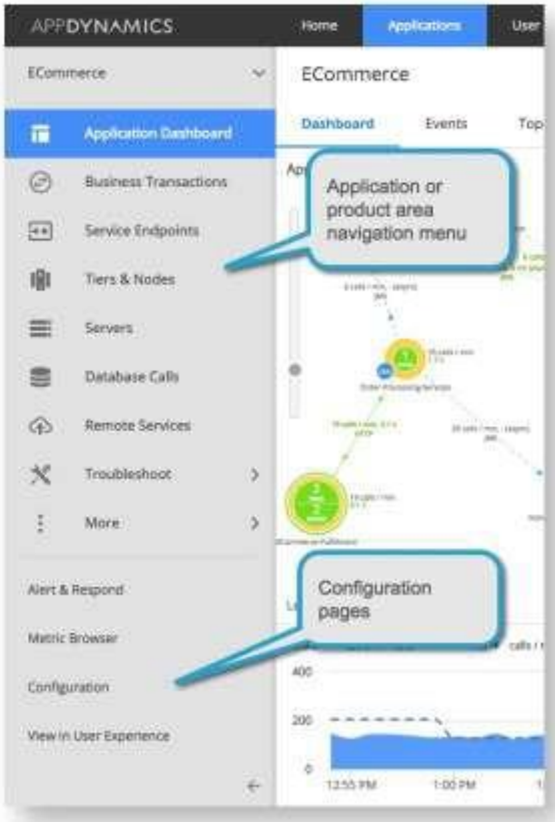
Top Navigation Bar

Use to navigate the major product areas and cross product feature areas, such as Alert & Respond and Dashboard & Reports.



Left Navigation Menu

Within a product area, a left navigation menu lets you access the various sections specific for that area.



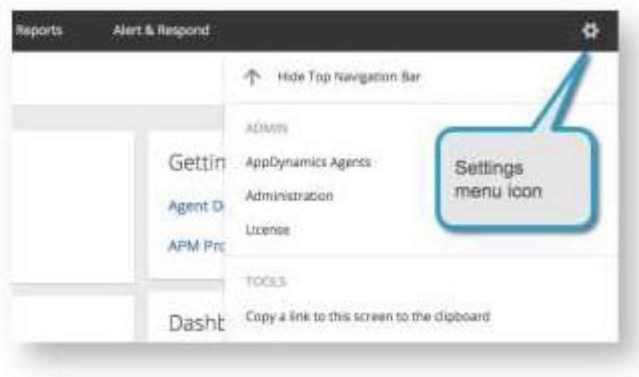
Application or product area navigation menu

Configuration pages

You can access configuration settings throughout the UI, but the Configuration link in the left menu (as called out above) serves as a consolidated point for accessing those settings in the UI. Configuration settings control various aspects of the behavior of AppDynamics, such as transaction and backend detection, baselines, and so on.

Settings Menu

Clicking the gear icon (⚙) shows the Settings menu. The Settings menu presents links for pages where you can customize the UI environment, view your account information, or, if permitted by your user role, configure administration settings for the UI such as add user accounts, configure authentication, and more.



Time Range Menu

Controls what metric data AppDynamics represents in the UI. The UI only shows the data generated in the selected time range.

The Time Range menu appears in dashboards, the Metric Browser, and elsewhere in the UI. The time range selection determines what data appears in the main panel throughout the UI, except in the case of pages such as the Scalability Analysis page, where other time contexts apply. When you choose a new time range from the menu, it remains the active range as you navigate to other pages in the UI.

The time range menu has preconfigured time ranges you can choose from, or you can enter a custom time range. You can set custom time ranges that terminate in the past, making them useful for analyzing trends over time.

You can save custom time ranges as a named range for viewing later or for sharing with teammates. To save a time range, after you have selected a custom time range, click the Save icon next to the menu. Once saved, your custom named time range appears in the time range menu. Remove, modify, or share saved time ranges from the Manage Custom Time Range option accessible from the Time Range menu.

For a flow map, you can choose to enable the auto-refresh functionality only for 5 minutes and 15 minutes time range. From the standard time range list, select 5 or 15 minutes to view the Auto-refresh checkbox at the bottom of the list. Selecting the Auto-refresh checkbox refreshes data automatically for the chosen standard time range.

Sharing Views With Other Users

Troubleshooting an application issue is often a collaborative effort involving multiple Controller UI users. You can share a particular view of the UI, such as a flow map or metric browser view, with other users using deep links.

A deep link is a shareable URL that retains the time range currently active in the UI. Time ranges that are relative to the current time, such as 'last 15 minutes', are represented as fixed time ranges in the URL.

To share a view, click **Settings** > **Copy a link to this screen to the clipboard**. You can then paste the URL where desired and share, for example in an email or chat window.

4 USING APPDYNAMICS GOVAPM

The following sections of the User Guide provide details of using AppDynamics GovAPM across the product services including Application Monitoring, End User Monitoring, Database Visibility, Infrastructure Visibility and Application Analytics.

4.1 APPDYNAMICS GOVAPM ESSENTIALS

This section introduces you to the AppDynamics GovAPM system. It describes concepts and procedures for the system that are common across the GovAPM product modules, from Application Performance Monitoring to Application Analytics, End User Monitoring, and more. It also covers the Controller UI, the browser-based console you use to understand, tune, and troubleshoot your application environment.

Controller UI Overview

The AppDynamics GovAPM Controller UI is the primary interface you will use to monitor, troubleshoot, and analyze your entire application landscape, from backend infrastructure and servers to the end user's app.
For additional information, see Controller UI Overview.

Sensitive Data Collection and Security

AppDynamics GovAPM collects data on the performance, health, and resources of an application, its components (transactions, code libraries), and the related infrastructure (nodes, tiers) that services those components. Before using AppDynamics, you should ensure that your application is secure and does not expose sensitive data.
For additional information, see Sensitive Data Collection and Security and Data Collection Dashboard.

Manager Users and Groups

Most Controller UI pages are access-controlled. You can add user accounts in the Controller UI, allowing other users to access the UI and configure AppDynamics GovAPM. The Account Owner is the predefined role with the Administer users, groups, and roles permissions.

For additional information, see Manage Users and Groups.

Metrics and Graphs

AppDynamics reports metrics for monitored systems over time.

The AppDynamics UI provides a variety of graphs and tools to help you visualize metric data so you can use it effectively for analyzing your application's performance.

For additional information, see Metrics and Graphs.

The Metric Browser is the main interface through which you inspect metrics. The Metric Browser displays all the metric values collected over the selected time period as an expandable tree.

For additional information, see Metric Browser.

Monitor Events

In the AppDynamics model, events represent a change in the state of a monitored application. An event can represent an error or exception generated by the application, the crossing of a performance threshold, or an operational change in the application, such as a JVM restart.

You can define policies that generate notifications or perform actions when an event occurs. See Alert and Respond for more information. You can also use events to investigate application issues underlying performance problems.
For additional information, see Monitor Events.

When investigating an application error or poor performance, you may want to see whether any application changes occurred at the same time as the error or issue.

AppDynamics automatically detects and reports common application change events, such as app server restarts or changes to application environment variables. You can supplement automatically detected change events with custom application change events, generated either through the UI or the REST API.
For additional information, see Monitor Application Change Events.

Monitor Infrastructure

While business transaction performance is the typical focus of a performance monitoring strategy, infrastructure performance can add insight into underlying factors in business transaction performance. Infrastructure Visibility provides visibility into the underlying OS infrastructure and networks on which your applications run. These agents enable you to isolate, identify, and troubleshoot infrastructure problems that can affect application performance such as resource-hogging tiers.

AppDynamics provides preconfigured application infrastructure metrics and default health rules to enable you to discover and correct infrastructure problems. You can also configure additional persistent metrics to implement a monitoring strategy specific to your business needs and application architecture.

In addition to health rules, you can view infrastructure metrics in the Metric Browser. Correlation Analysis and Scalability Analysis charts in the browser can be particularly useful for understanding how infrastructure metrics correlate to business transaction performance.

For more information, see Overview of Infrastructure Visibility.

Alert and Respond

AppDynamics can generate notifications or take other types of actions based on conditions or events you configure. Using the alert and respond feature, you can find out about problems as they happen, or even before they happen when you define alerts on warning conditions.

For additional information, see Alert and Respond.

Dashboards and Reports

Dashboards and Reports are useful for getting a high-level overview of the health and performance of your system on a regular basis.

Dashboards provide a graphical overview of the selected data made available for quick access. Custom dashboards let you create and arrange widgets to give users a visual overview of the data of interest to them. A War Room is similar to a dashboard, but it is interactive. The chat facility in virtual war rooms enables visually-enhanced collaborative troubleshooting in real time in response to high priority events.

Reports capture AppDynamics data from dashboards and help share it with the recipients. Reports can be automatically generated on a regular schedule.

- Virtual War Rooms
- Widgets
- Custom Dashboards
- Reports

For additional information, see Dashboards and Reports.

4.2 APPLICATION MONITORING

AppDynamics Application Performance Monitoring (APM), a component of the AppDynamics GovAPM system, gives you end-to-end visibility into the performance of your applications.

AppDynamics works with popular programming languages such as Java, .NET, Node.js, PHP, Python, C/C++, and more, enabling you to:

- Troubleshoot problems such as slow response times and application errors.
- Automatically discover application topology and how components in the application environment work together to fulfill key business transactions for its users.
- Measure end-to-end business transaction performance, along with the health of individual application and infrastructure nodes.
- Receive alerts based on custom or built-in health rules, including rules against dynamic performance baselines that alert you to issues in the context of business transactions.
- Analyze your applications at the code execution level using snapshots.

Application Model Overview

This is described in **Section 2**.

Install App Server Agents

Instrumenting an application adds the AppDynamics application agent (app agent) into the runtime process of the application. This section gives you an overview of installing agents in the application environment.

Agent Installation Quick Start

The Getting Started Wizard in the AppDynamics GovAPM Controller walks you through the steps to download and configure an agent for your application. This section gives you an overview of how to use the wizard.

The wizard produces a fully configured agent, including a node identity. Accordingly, it is intended to be run for each application instance you want to monitor until you have the hang customizing the configuration manually.

Before Starting

Before installing an agent, make sure that:

The agent supports your application environment. See App Server Supported Environments.

You can access the application host with a user account that has sufficient privileges to install the agent and—for certain installation types—restart the application.

The application host has network connectivity to the Controller. Proxies or firewalls on the network between the agent and Controller may require additional configuration.

Use the Getting Started Wizard

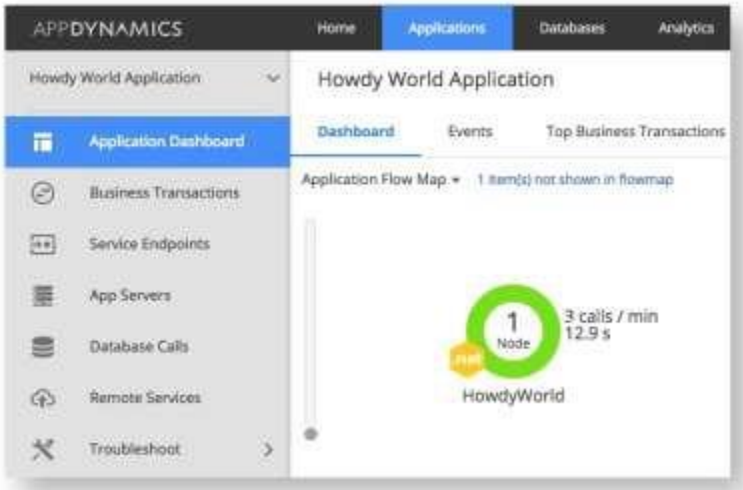
After verifying the requirements, follow the workflow as guided by the wizard:

Open the wizard from the home page in the controller UI by clicking Getting Started.

Enter the configuration values for the application instance as described in the wizard. You name

the node and indicate the tier and business application to which it will belong. If you are not sure of the best values to use, you can use temporary names and change them later.
When finished, download the agent. How you download the agent varies by agent type. In most cases, the agent comes as a ZIP file that you extract and install in the startup routine of your server. For other types of agents, you may need to modify the instrumented source code, for example, by including the agent library. The wizard walks you through it for each agent type.
Install the agent on your app server.

Apply load to your application. If you are instrumenting a production application, this will happen with customer interaction. Otherwise, create some test load on your application. The agent instruments the application code and reports metrics back to the Controller.
View your application in the Application Dashboard. For example:



The wizard makes it easy to perform a basic installation of the agent with minimally required settings, such as the Controller host and port, SSL, application name, and tier name. For advanced options or more complicated scenarios, you need to perform a manual installation of the agent. For more information, see the agent-specific link in the following section.

Agent Installation by Type

For detailed installation information by agent type, see the following topics:

- Java Agent
- .NET Agent
- Node.js Agent
- PHP Agent
- Python Agent
- Apache Web Server Agent
- C/C++ Application Agent
- Go Language Agent
- Standalone Machine Agents

For more information, see Install App Server Agents.

Administer App Server Agents

For detailed information, see Administer App Server Agents.

Business Applications

A business application serves as the top-level container in the AppDynamics APM model. Business applications contain a set of related business transactions and service endpoints, along with the infrastructure components and artifacts that interact to provide the services.

You can model your environment into one or more business applications. When you have multiple business applications within the same AppDynamics account, you can use the cross application flow map to view relationships between the different applications. If you have a more complex environment with multiple AppDynamics accounts, you can establish federated friendship between the accounts to view the federated cross application flow.

For additional information, see Business Applications.

Business Transactions

In the AppDynamics application model, a business transaction represents the end-to-end, cross-tier processing path used to fulfill a request for a service provided by the application. This topic introduces and describes business transactions.

For additional information, see Business Transactions.

Service Endpoints

A business transaction offers a view of application performance that cuts across the environment, reflecting all services that participate in fulfilling the transaction. Another way of viewing application performance, however, focuses on the performance of a particular service independently of the business transactions that use it.

Like business transactions, service endpoints give you key performance indicators, metrics, and snapshots. However, they provide that information exclusively in the context of that service, omitting business transaction context or downstream performance data.

For additional information, see Service Endpoints.

Tiers and Nodes

The Tier list in the Controller UI lets you view and organize the tiers in your business application and the monitored nodes within each tier.

For additional information, see Tiers and Nodes.

Remote Services

This topic describes the monitoring information available for remote services, also known as backends.

For additional information, see Remote Services.

Troubleshooting Applications

AppDynamics includes many tools and views designed to help you diagnose application-related problems, as introduced in this section.

For additional information, see Troubleshooting Applications.

Information Points

You can use information points to define custom metrics based on methods you configure.

Information points are similar to data collectors. However, while data collectors show application data in the context of a business transaction, information points reflect data state across all invocations of a method, independently of business transactions. They also let you apply computations to the values, for example, representing the sum or average for a method return value or input parameter.

This topic describes how to configure and use information points.
For additional information, see Information Points.

Configure Instrumentation

In AppDynamics, instrumentation refers to how app agents interact with your application software to gather performance data and report it back to the AppDynamics Controller. The first step to instrumenting your application is installing the app agents on the servers where the code runs.

App agents ship with default instrumentation settings that cover the most common types of application frameworks and programming patterns. If your requirements are more complex, you can customize the instrumentation behavior to suit your environment.

For additional information, see Configure Instrumentation.

Development Level Monitoring

Development level monitoring is a mode in which certain default limits on the data that AppDynamics collects are disabled, giving you additional visibility on application activities. This topic describes development level monitoring.

For additional information, see Development Level Monitoring.

App Server Agents Supported Environments

This topic provides an aggregated view of the supported environments for the app server agents.
For additional information, see App Server Agents Supported Environments.

4.3 END USER MONITORING

AppDynamics GovAPM End User Monitoring (EUM) gives you end-to-end visibility on the performance of your web applications. EUM helps you troubleshoot problems such as slow web or Ajax. EUM provides metrics on application performance and user activity, such as:

- How server performance impacts your web and device performance
- How third-party APIs impact your web and device performance
- Where your heaviest loads originate
- How your users connect to and navigate your application

Overview of End User Monitoring

AppDynamics End User Monitoring (EUM) gives you visibility on the performance of your application from the viewpoint of the end user.

While Application Performance Monitoring (APM) measures user interaction starting at the web server or application server entry point, EUM extends that visibility all the way to the web browser. As a result, EUM reveals the impact the network and browser rendering time have on the user experience of your application.

Understand End User Activity

- Using EUM, you can determine:
- Where geographically your heaviest application load is originated.
- Where geographically your slowest End User response times occur.
- How performance varies by
  - location
  - client type, device, browser and browser version, and network connection for web requests
  - application and application version, operating system version, and device
- What your slowest web/Ajax requests are, and where the problem may lie.
- How application server performance impacts the performance of your web traffic.

View EUM Data

The performance information generated by EUM is distinct from the application monitoring data generated by app server agents.

EUM data appears in various locations in the Controller UI, including in the User Experience dashboard, Metric Browser, and AppDynamics Analytics pages.

When linked to application business transactions, EUM data gives you a complete view of your end users' experience from the client request, through the application environment, and back to the client as the user response.

You can view EUM performance data in the Controller UI in the User Experience tab. From there, you can access information specific to browser applications.

For additional information, see Overview of End User Monitoring.

EUM Data

EUM collects, aggregates, and stores many different types of data that are displayed in different UI components of the Controller.

This topic is to help you understand the following:

- the types of EUM data
- how data is collected
- where data is stored
- where data is displayed in the Controller

For additional information, see EUM Data.

Browser Monitoring

AppDynamics Browser Monitoring provides two products that can help you improve your user's experience: Browser Real User Monitoring (Browser RUM) and Browser Synthetic Monitoring. Browser RUM allows you to see how your web application is performing, using real user data to analyze application performance and user experience. Browser Synthetic Monitoring allows you to analyze application availability and performance, using scheduled testing to analyze website availability. This topic provides an overview of Browser RUM and Browser Synthetic Monitoring, the differences between the two products, and how to use them together to monitor your browser applications.

For additional information, see Browser Monitoring.

4.4 DATABASE VISIBILITY

AppDynamics Database Visibility provides end-to-end visibility on the performance of your database. Database Visibility helps you troubleshoot problems such as slow response times and excessive load. Database Visibility provides metrics on database activity, such as:

- SQL statements or stored procedures that are consuming most of the system resources
- Statistics on procedures, SQL statements, and SQL query plans
- Time spent on fetching, sorting, or waiting on a lock
- Activity from the previous day, week, or month

Overview of Database Visibility

AppDynamics Database Visibility consists of four main components:

- Database Agent
- Collector
- Controller
- Events Service

Database Agent

The AppDynamics Database Agent is a standalone Java program that collects performance metrics about your database instances and database servers.

You can deploy the Database Agent on any machine running Java 1.8 or higher. The machine must have network access to the AppDynamics Controller and the database instance that you want to be monitored.

A database agent running on a typical machine with 16 GB of memory can monitor about 25 databases. On larger machines, a database agent can monitor up to 200 databases. To monitor more than 100 databases, increase the initially allocated heap size for the Database Agent JVM. See Database Visibility Supported Environments

Collector

The Database Agent Collector is the process that runs within the Database Agent to collect performance metrics about your database instances and database servers. One Collector collects metrics for one database instance. Multiple collectors can run in one Database Agent.

AppDynamics Database Visibility detects when a database back end has matching credentials with a database server being monitored by a Collector. Database Visibility automatically associates the back end with the collector, so you can view its performance with the Application Flow Map, Tier Flow Map, or Node Flow Map.

Controller

The Controller is the central interface where you can see all your database instance and database server performance metrics.

The following types of information are sent to the Controller:

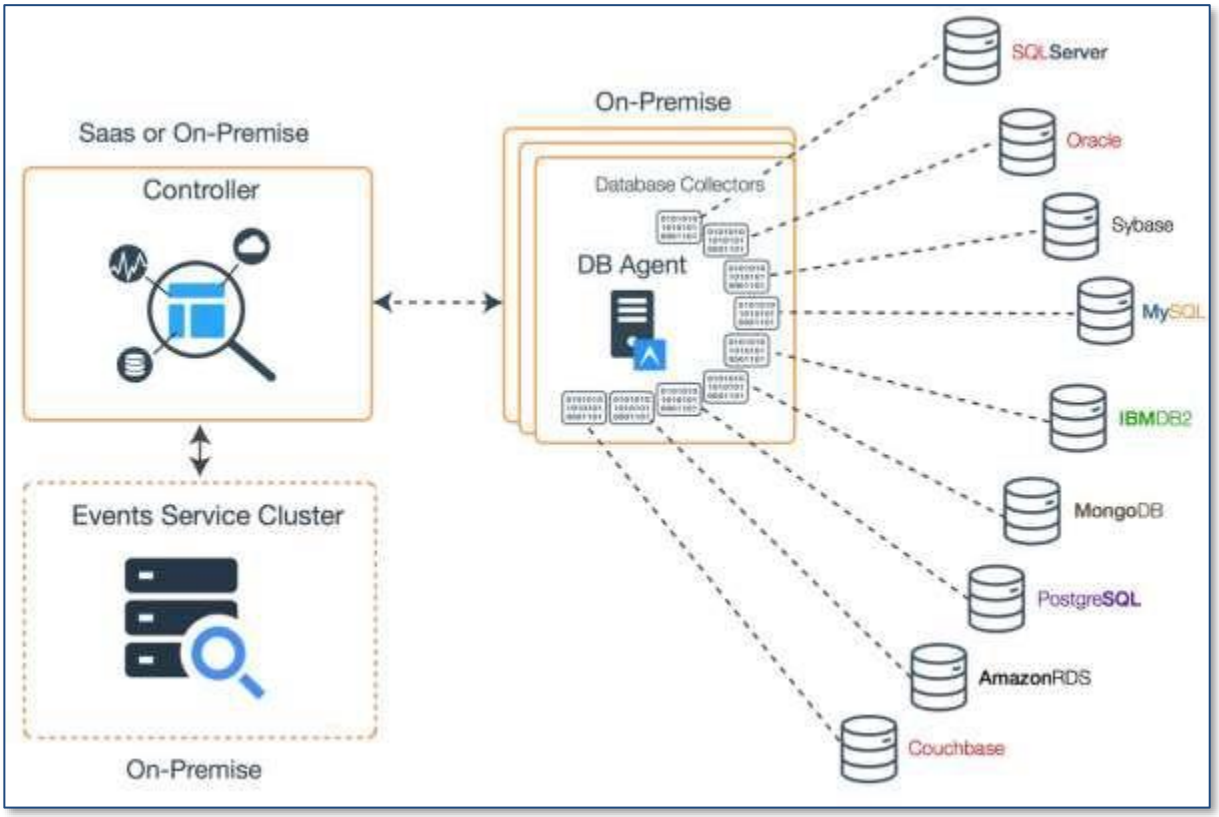
- Database-level metrics, such as number of queries processed and other database statistics
- Names and attributes of all sessions, clients, queries, and other objects on the monitored system

Events Service

Event service stores high volumes of metric data.

The following types of information are sent to the Events Service:

- time that each query spends at each wait state
- individual query statistics for databases that support it
- information about individual execution plans in databases that support it



For additional information, see Overview of Database Visibility.

Database Visibility Supported Environments

This topic discusses application environments and versions that are supported by the Database Visibility Agent.

For additional information, see Database Visibility Supported Environments.

Database Visibility System Requirements

This topic lists hardware and software requirements using Database Visibility.
For additional information, see Database Visibility System Requirements.

Administer the Database Agent

The Database Agent is a standalone Java program that collects performance metrics about your database instances and database servers. You can view these performance metrics in the Metric Browser of the AppDynamics Controller UI.
To start monitoring your database, you must first install the Database Agent. You may want to install multiple agents for the following reasons:

- You have databases on multiple networks that are not accessible by one machine.
- You have multiple SQL Server instances that each require different credentials, and you want to connect to them through Windows Authentication.
- You want one or more agents to back up a primary agent. Backup agents ensure that your database instances are continually monitored during agent failure or machine downtime.

For additional information, see Administer the Database Agent.

Add Database Collectors

For additional information, see Add Database Collectors.

Monitor Databases and Database Servers

The topics in this section show you how to use the features of AppDynamics to monitor and troubleshoot your database environments.

For additional information, see Monitor Databases and Database Servers.

4.5 INFRASTRUCTURE VISIBILITY

AppDynamics Infrastructure Visibility gives you end-to-end visibility on the performance of the hardware running your applications. Infrastructure Visibility helps you identify and troubleshoot problems that can affect application performance such as server failures, JVM crashes, and network packet loss. Infrastructure Visibility provides metrics such as:

- CPU busy/idle times, disk and partition reads/writes, and network interface utilization (Machine Agents)
- Packet loss, round trip times, connection setup/teardown errors, TCP window size issues, and retransmission timeouts (Network Visibility, additional license required)
- Disk/CPU/memory utilization, process, and machine availability (Server Visibility, additional license required)

Overview of Infrastructure Visibility

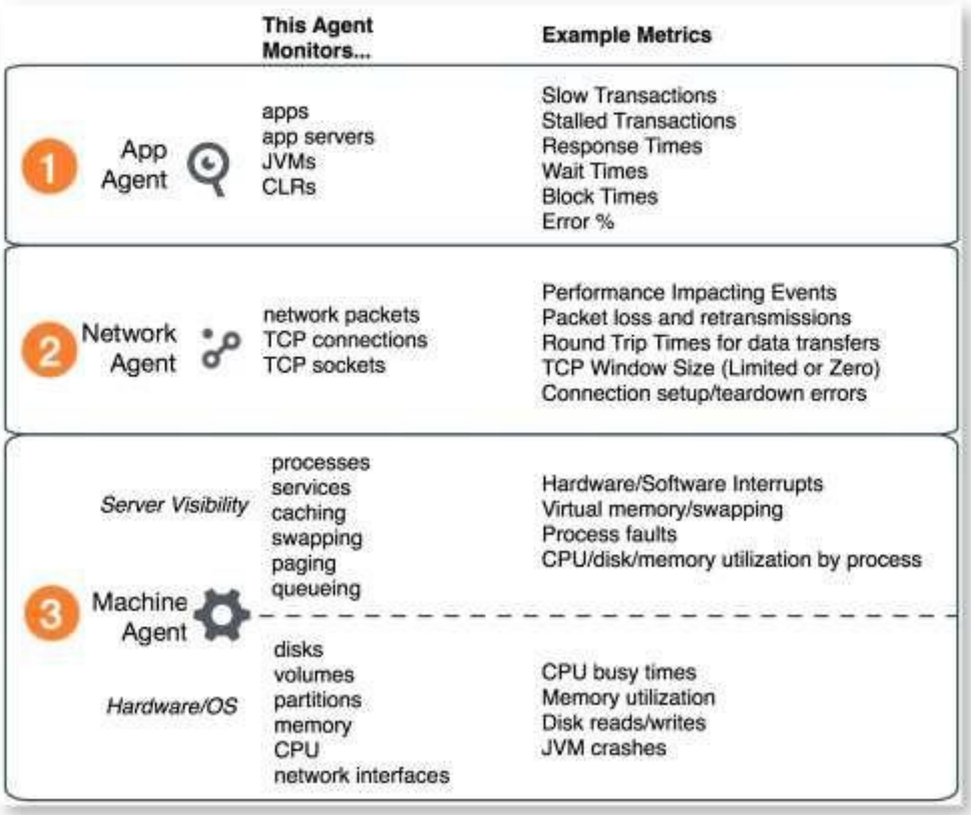
Many times the root cause of application issues is most obvious by looking at application, network, server, and machine metrics that measure infrastructure utilization. For example, the following infrastructure issues can slow down your application:

- Too much time spent in garbage collection of temporary objects (application metric)
- Packet loss between two nodes that results in retransmissions and slow calls (network metric)
- Inefficient processes that result in high CPU utilization (server metric)
- Excessively high rates of reads/writes on a specific disk or partition (hardware metric)

Infrastructure Visibility enables you to isolate, identify, and troubleshoot these types of issues. Infrastructure Visibility is based on a machine agent that runs with an app server agent on the same machine. These two agents provide multi-layer monitoring as follows:

1. The app server agent collects metrics about applications and identifies applications, tiers, and nodes with slow transactions, stalled transactions, and other application-performance issues.
2. The Network Agent monitors the network packets sent and received on each node and identifies lost/retransmitted packets, TCP bottlenecks, high round trip times, and other network issues.
3. The machine agent collects metrics at two levels:
    a. Server Visibility metrics about local processes, services, and resource utilization.
    b. Basic machine metrics about disks, memory, CPU, and network interfaces.

This multi-layer monitoring enables you to find possible correlations between application issues and service, process, hardware, network, or other issues on the machine.



		This Agent Monitors...	Example Metrics
1 App Agent		apps app servers JVMs CLRs	Slow Transactions Stalled Transactions Response Times Wait Times Block Times Error %
2 Network Agent		network packets TCP connections TCP sockets	Performance Impacting Events Packet loss and retransmissions Round Trip Times for data transfers TCP Window Size (Limited or Zero) Connection setup/teardown errors
3 Machine Agent	*Server Visibility*	processes services caching swapping paging queueing	Hardware/Software Interrupts Virtual memory/swapping Process faults CPU/disk/memory utilization by process
	Hardware/OS	disks volumes partitions memory CPU network interfaces	CPU busy times Memory utilization Disk reads/writes JVM crashes

For additional information, see Overview of Infrastructure Visibility.

Hardware Resources Metrics

This topic describes the basic hardware metrics collected by the Machine Agent and the additional metrics collected by the Standalone Machine Agent for Server Visibility. Not all of the metrics that appear in the Hardware Resources branch of the Metric Browser tree are provided by the Machine Agent; some are collected by the Database Agent and are used in the

Database Monitoring UI, while others might be custom metrics added by another extension run by your Standalone Machine Agent.

For additional information, see Hardware Resources Metrics.

Standalone Machine Agent

You use the Standalone Machine Agent (Machine Agent) to collect basic hardware metrics. The Machine Agent is a Java program that has an extensible architecture enabling you to supplement the basic metrics reported in the AppDynamics Controller UI with your own custom metrics. The Machine Agent is also the delivery vehicle for AppDynamics Server Visibility, which provides an expanded set of hardware metrics and additional monitoring capability. .

For additional information, see Standalone Machine Agent.

Server Visibility

The AppDynamics Server Visibility module uses the Machine Agent to provide extended hardware metrics and Service Availability Monitoring. Server Visibility includes additional windows in the Controller UI that enable you to quickly see underlying infrastructure issues impacting your application performance. This means you can rapidly troubleshoot hardware performance problems that are impacting your applications.

For additional information, see Server Visibility.

Network Visibility

With Network Visibility, you can

- Answer the key question quickly and easily: *Is the network to blame for any performance issues in my application?*
- Identify the root cause of performance issues in the application, the network, or in application/network interactions (i.e., how an application or server utilizes the network).
- Pinpoint the traffic flows, individual nodes, and transport connections where network or application/network issues are occurring.
- Collect detailed metrics that show how the network is performing and how well your application uses network connections and resources.
- Collect and report targeted troubleshooting information to network, IT, DevOps, and other teams.

For additional information, see Network Visibility.

Monitoring Containers

You can use the Machine Agent to monitor application nodes running inside Docker containers and to identify container issues that impact application performance. By viewing and comparing

APM metrics with the underlying container and server/machine metrics, you can easily answer the question: Is my application problem purely an application problem, or is the root cause in the container or the server?

For additional information, see Monitoring Containers.

Monitoring Kubernetes

With AppDynamics you can gain real-time visibility into your containerized applications deployed to Kubernetes. Kubernetes is an open-source container-orchestration platform for automating deployment, scaling, and management of applications running in containers.

For additional information, see Monitoring Kubernetes.

Container Metrics

The Containers Dashboard shows resource utilization metrics of your containerized applications and the underlying host machine. You can access the Containers Dashboard from the Servers tab in the Controller UI. The Containers page lists all monitored containers used by an application that is registered to the Controller.

For additional information, see Container Metrics.

4.6 APPLICATION ANALYTICS

Integration of all AppDynamics Application Performance Management (APM) performance engines results in a dynamic, accessible set of data about every business transaction. Application Analytics can extract that information and generate baselines and dashboards, and give you a perspective beyond traditional APM, enabling you to do real-time analysis of business performance that is correlated with the performance of your application software. You can use Application Analytics with the APM, Browser RUM, and Browser Synthetic Monitoring product modules to generate:

- Transaction Analytics
- Log Analytics
- Browser Analytics

Overview of Application Analytics

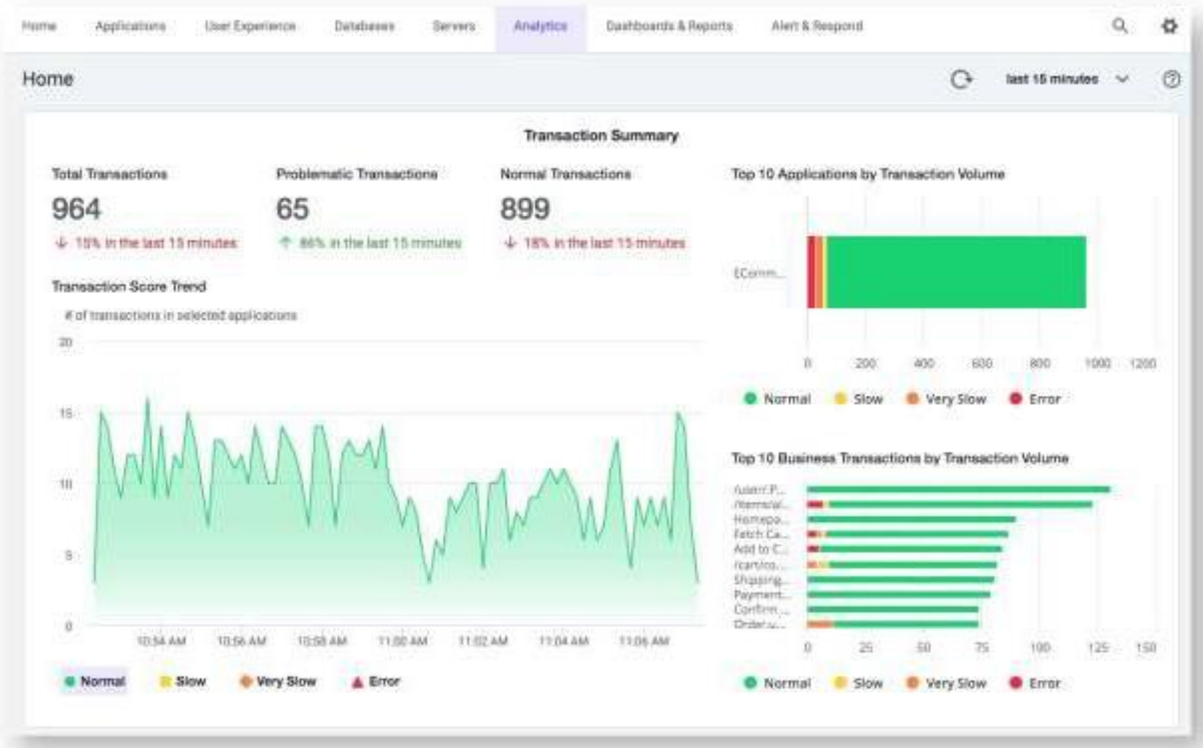
Application Analytics can help you answer business-oriented questions, such as:

- How many users experienced failed checkout transactions in the last 24 hours?
- How much revenue was lost because of these failures? How is that revenue distributed across different product categories?
- What is your revenue for the day for a geographical region?
- What was the revenue impact by product category associated with two marketing campaigns we ran last week?

Analytics Home Page

The Analytics Home page consolidates data from transaction, and browser events. The home page automatically generates Transaction and End User Monitoring Summary panels through queries that aggregate data into widgets.

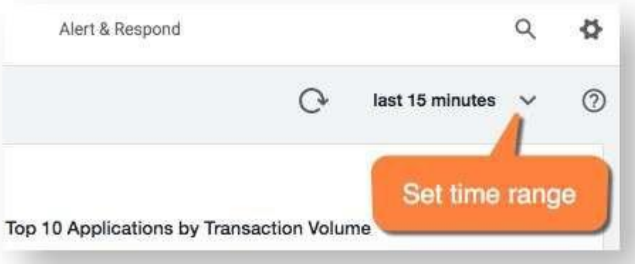
You can access the home page at any time by clicking the **Home** icon on the left navigation pane in Analytics.



Home Page Data

AppDynamics Query Language (ADQL) queries generate the data for each home page widget. You cannot access or edit the ADQL queries.

Your selected time-range automatically applies a time filter to each query. Widgets refresh every two minutes.



Transaction Summary

The **Transaction Summary** panel aggregates your Analytics data from applications and business transactions. **Transaction Summary** widgets are categorized by transaction threshold, from normal to error. The **Transaction Summary** panel contains six widgets:

- **Total Transactions -** count of all transactions
- **Problematic Transactions -** count of transactions that are not within normal threshold boundaries
- **Normal Transactions -** count of transactions that fall within the normal threshold boundaries**Top 10 Applications by Transaction Volume** - graphical view of applications
- **Top 10 Business Transactions by Transaction Volume** - graphical view of transactions
- **Transaction Score Trend** - total number of transactions for all applications

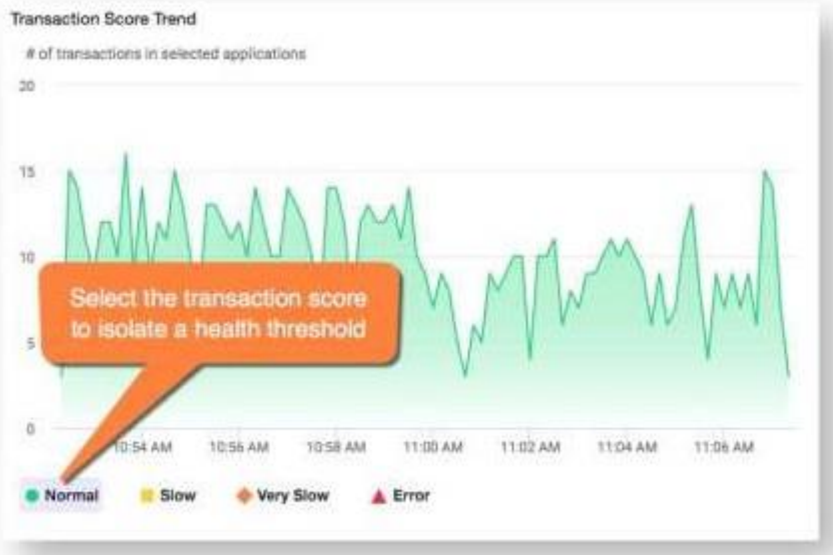
End User Session Summary

The **End User Session Summary** panel pulls together browser session data into widgets. The **End User Session Summary** panel contains six widgets:

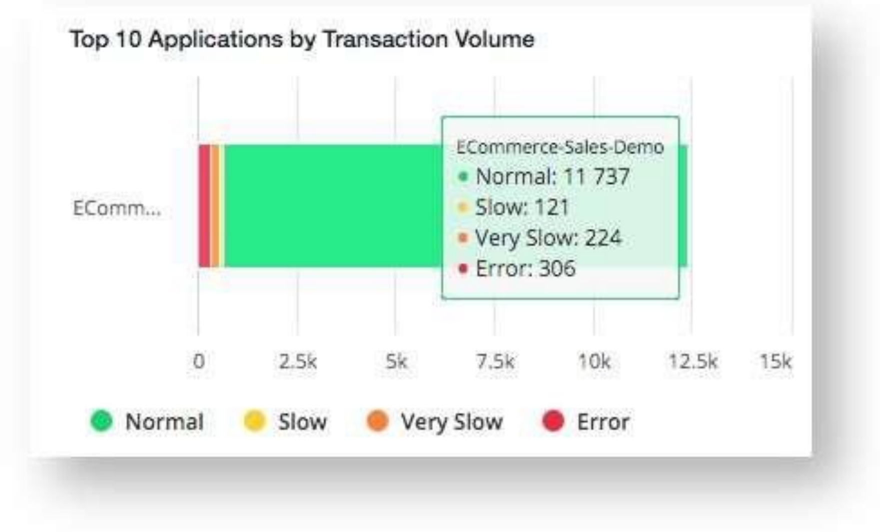
- **Browser Sessions -** count of unique browser sessions
- **Browser Users -** count of browser users
- **Browser Session Bounce Rate** - rate of bounces per total sessions
  A bounce is a browser session where the page count equals one. Bounces occur when a user lands on a webpage and leaves the site without visiting additional pages.
- **Top Page Referrers by Session Volume** - the website visited by your user before accessing your site
  Referrer is an optional component of the Events Service. If you have not configured referrers, this widget does not contain data.
- **Session Trend** - total number of sessions separated by browser sessions

Using Home Page Data

You can select the transaction score in each chart to isolate metrics. For example, hovering over **Normal** in **Transaction Score Trend** removes other transaction scores, as demonstrated below.



In graph widgets, you can hover over any point to see a count of the data, as shown below.



RBAC

Analytics enforces role-based access control (RBAC). As a result of RBAC, your Analytics Home page displays only the data to which you have access.

Analytics Data Sources

Application Analytics can collect data from many sources in your AppDynamics GovAPM system, as listed in this topic.

For additional information, see Analytics Data Sources.

Administer the Analytics Agent

Application Analytics is built on the AppDynamics GovAPM Application Performance Management (APM), which includes the Events Service, the unstructured document store for the platform. The exact components you need to install depend upon your environment and your requirements.

For additional information, see Administer the Analytics Agent.

Configuring Application Analytics

To configure the system to collect the data in which you are interested, see Configuring Application Analytics.

Using Application Analytics Data

This topic provides an overview of how to use the Application Analytics data. You have several strategies available for locating and using analytics data of interest. Once you have identified data that is useful for your business, you can use visualization or other features to represent and monitor that data.

For additional information, see Using Application Analytics Data.

ADQL Reference

The AppDynamics Query Language (ADQL) is a query language for searching data available in Application Analytics. ADQL is similar to the **SELECT** statement in the widely used Structured Query Language (SQL) but is designed specifically for AppDynamics Analytics data. Like the **SELECT** command in SQL, ADQL enables you to specify the data source (such as transaction events, EUM events, or logs), a list of fields to retrieve, and conditions for selecting events from the analytics data store.

For additional information, see ADQL Reference.

APPENDIX A: GLOSSARY

The table below presents definitions for terms used in this User Guide.

TERM	MEANING
Action	An automatic response to an event based on a policy. Examples include sending alerts, taking diagnostic snapshots, remediation through scripts, or making a REST API call to integrate with other tools. Actions are customizable.
Agent	Code that collects and reports data. Agents can refer to languages (Java, Node.js, Python, etc.), machines (hardware and network data), and databases, etc.
Alert	Email, SMS, or customized external notification interface that notifies you of a problem or event.
Application	A named collection of tiers representing a monitored environment. Also referred to as Business Application.
Backend	Databases, remote services, and any detected out of process components that are involved in business transaction processing.
Business Transaction	An action of your application. Business transactions represent your critical business functions. Examples include login, browse, and checkout. A business transaction represents the end-to-end, cross-tier processing path used to fulfill a request for a service provided by the application.
Controller	Collects, stores, analyzes, and baselines performance data collected by agents.
Data Collector	Enables you to collect supplemental application data for business transaction snapshots and/or analytics. There are three types of data collectors: Method Invocation (MIDC), HTTP, and SQL.
Event	Unstructured data that represents an action or occurrence detected by the agent. Examples include JVM restart, deadlock, and error.
Flow Map	Graphical view of information flow within a business application. Flow maps show the tiers, nodes, message queues, and databases in the environment, and the business transactions that flow through them.

TERM	MEANING
Health Rule	Allows you to select specific metrics as key to the overall health of an application and to define ranges for acceptable performance of those metrics. You can customize default health rules or create new ones.
Information Point	Instruments a method in application code outside the context of any business transaction. Used for monitoring the performance of the method itself, or for capturing numerical data from the method's input parameters or return value.
Metric	Numerical measurement over a time interval. A data point is a numerical value that an agent measures and reports. Examples include "calls per minute" and "average response time."
Node	The smallest unit of the AppDynamics modeling environment.
Policy	Provides a mechanism to automate monitoring, alerting, and problem remediation. Consists of: A trigger based on events An action in response to the trigger
Service Endpoint	Provides basic metrics for one application service, or group of services, provided by a tier.
Tier	A named collection of nodes, typically representing a service. A tier is a unit in the AppDynamics model composed of a grouping of one or more nodes.
Transaction Snapshot	A collection of detailed diagnostic information captured during a single invocation of a business transaction. Examples include call graphs and timing during a single user login, where login is the business transaction.