Microsoft

AZ-204T00A

Learning Path 11: Troubleshoot solutions by using Application Insights

Agenda

- Monitor app performance

Module 1: Monitor app performance

Learning objectives

- Describe how Application Insights works and how it collects events and metrics.
- Instrument an app for monitoring, perform availability tests, and use Application Map to help you monitor performance and troubleshoot issues.

Introduction

- Instrumenting and monitoring, your apps helps you maximize their availability and performance.
- Application Insights is an extension of Azure Monitor and provides Application Performance Monitoring (also known as “APM”) features.
- In addition to collecting metrics and application telemetry data, which describe application activities and health, Application Insights can also be used to collect and store application trace logging data.

Explore Application Insights (1 of 2)

Features include, but not limited to:

Feature	Description
Live Metrics	Observe activity from your deployed application in real time with no effect on the host environment.
Availability	Also known as “Synthetic Transaction Monitoring”, probe your applications external endpoint(s) to test the overall availability and responsiveness over time.
GitHub/DevOps integration	Create GitHub or Azure DevOps work items in context of Application Insights data.
Usage	Understand which features are popular with users and how users interact and use your application
Smart Detection	Automatic failure and anomaly detection through proactive telemetry analysis.
Application Map	A high-level top-down view of the application architecture and at-a-glance visual references to component health and responsiveness.
Distributed Tracing	Search and visualize an end-to-end flow of a given execution or transaction.

Explore Application Insights (2 of 2)

Application Insights monitors:

- Request rates, response times, and failure rates
- Dependency rates, response times, and failure rates
- Exceptions
- Page views and load performance
- AJAX calls from web pages
- User and session counts.
- Performance counters
- Host diagnostics
- Diagnostic trace logs
- Custom events and metrics

Several ways to get started monitoring and analyzing performance:

- At run time
- At development time
- Instrument your web pages
- Analyze mobile app usage
- Availability tests

Discover log-based metrics

Log-based metrics

- The Application Insights backend stores all collected events as logs.
- The Application Insights blades in the Azure portal act as an analytical and diagnostic tool for visualizing event-based data from logs.
- Using logs to retain a complete set of events can bring great analytical and diagnostic value.
- Collecting a complete set of events may be impractical (or even impossible) for applications that generate a large volume of telemetry.

Pre-aggregated metrics

- Stored as pre-aggregated time series, and only with key dimensions
- The newer SDKs (Application Insights 2.7 SDK or later for .NET) pre-aggregate metrics during collection
- For the SDKs that don't implement pre-aggregation, the Application Insights backend still populates the new metrics by aggregating the events received by the Application Insights event collection endpoint

Instrument an app for monitoring

Auto-instrumentation

- Auto-instrumentation allows you to enable application monitoring with Application Insights without changing your code.
- Just enable and, in some cases, configure the agent, which will collect the telemetry automatically.
- You'll see the metrics, requests, and dependencies in your Application Insights resource

Instrumenting for distributed tracing

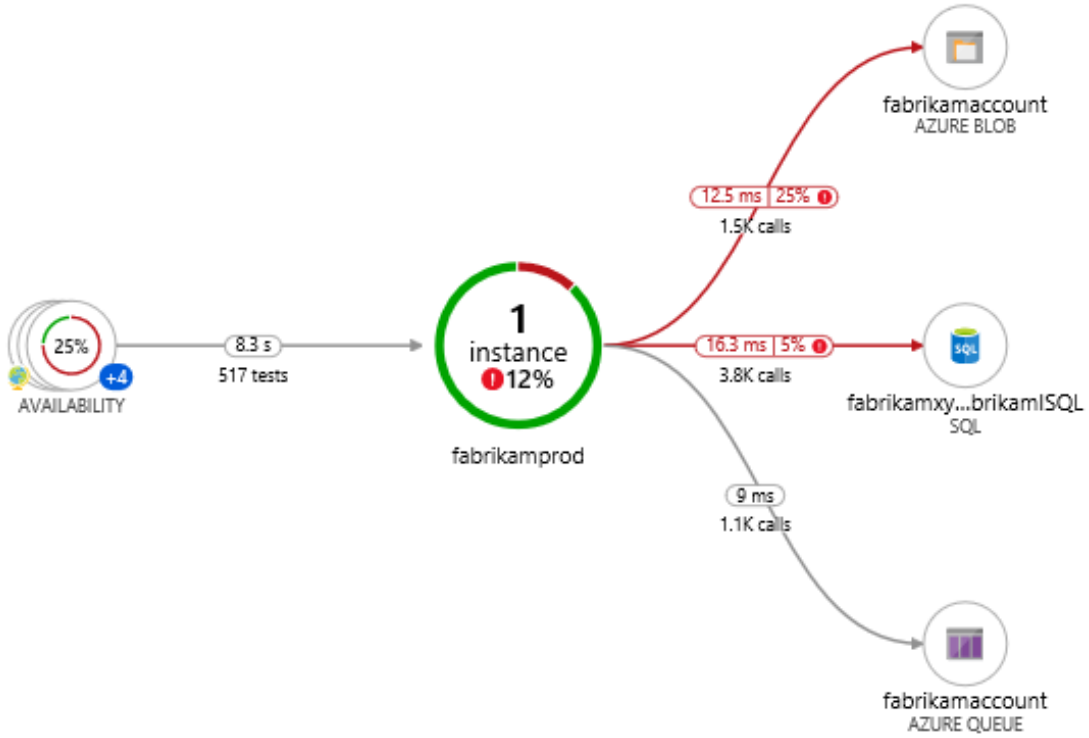
- How to Enable Distributed Tracing
- Enabling via Application Insights SDKs
- Enable via OpenCensus

Select an availability test

- You can set up availability tests for any HTTP or HTTPS endpoint
- You don't have to make any changes to the website you're testing
- It doesn't even have to be a site that you own, you can test the availability of a REST API that your service depends on.
- You can create up to 100 availability tests per Application Insights resource, and there are three types of availability tests
  - Standard test
  - Custom TrackAvailability

Troubleshoot app performance by using Application Map

- Application Map helps you spot performance bottlenecks or failure hotspots across all components of your distributed application.
- Each node on the map represents an application component or its dependencies; and has health KPI and alerts status.
- You can click through from any component to more detailed diagnostics
- Components are independently deployable parts of your distributed/microservices application



Summary and knowledge check

In this module, you learned how to:

- Describe how Application Insights works and how it collects events and metrics.
- Instrument an app for monitoring, perform availability tests, and use Application Map to help you monitor performance and troubleshoot issues.

1. What availability test is recommended for authentication tests?

2. What metric collection type provides near real-time querying and alerting on dimensions of metrics, and more responsive dashboards?

Discussion and lab

Group discussion questions

- Can you describe how Application Insights and Log Analytics are related to each other?
- What are the performance differences between log-based and pre-aggregated metrics? When would you choose one over the other?
- What process would you follow, and what tools would you choose, to monitor and optimize the performance of an app?

Lab 11: Monitor services that are deployed to Azure

In this lab, you will create an Application Insights resource in Azure that will be used to monitor and log application insight data for later review. The API will be set to automatically scale if demand increases to a certain threshold and logging the data will help determine how the service is being utilized.

http://aka.ms/az204labs

- Exercise 1: Create and configure Azure resources
- Exercise 2: Monitor a local web API by using Application Insights
- Exercise 3: Monitor a web API using Application Insights

End of presentation