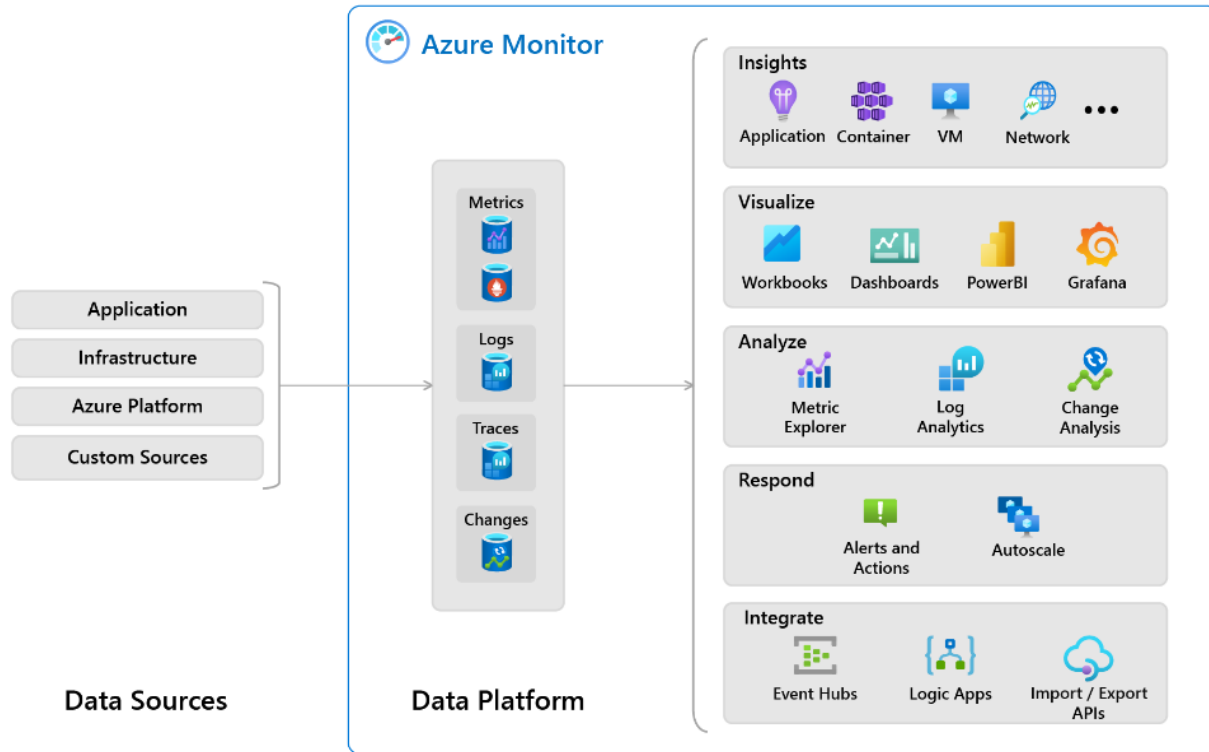


Instrument solutions to support monitoring and logging

Azure Monitor

Сервіс дозволяє збирати та аналізувати телеметрію.



В цей список входять:

- Metrics
- Logs
- Distributed traces
- Changes (збираються зміни в Azure ресурсах)

Insights and curated visualizations

Моніторинг повинен допомагати розуміти стан системи. Деякі Azure ресурси мають власні кастомні метрики та візуалізацію. Цей процес потребує мінімального налаштування. Також є механізм Insights, який дозволяє візуалізувати більш складні метрики для apps, containers, VMs.

Application Insights - моніторить доступність, продуктивність та навантаження web програм. По метрикам можна знаходити помилки ще до того, як користувачі їх зарепортять

Container Insights - все те ж саме, але метрики збираються з контейнерів, які запущені в Kubernetes. Так же збираються логи контейнерів.

VM Insights - моніторить віртуальні машини. Вміє ідентифікувати конкретні процеси та знаходити незаплановані залежності від зовнішніх процесів чи ресурсів.

Application insights

Можливості:

- Live Metrics - моніторинг програми в реальному часі без впливу на хост-середовище
- Availability - можна налаштувати health checks (потрібні окремі endpoints)
- GitHub or Azure DevOps integration - можна створювати таски (work items) прив'язуючи їх до даних з Azure Insights
- Usage - відображає навантаження, що дозволяє збирати статистику використання юзерами тих чи інших функцій системи
- Smart Detection - під час проактивного аналізу дозволяє знаходити аномалії чи падіння
- Application Map - є діаграма, яка відображає стан всіх сервісів (їх інстансів) та комунікацію між ними
- Distributed Tracing

What Application Insights monitors

- Request rates, response times, failure rates
- Dependency rates, response times, failure rates
- Exceptions
- Page views and load performance (метрики репортить браузер юзера)
- AJAX calls from web pages
- User and session counts
- Performance counters (CPU, Memory, Network)
- Host diagnostic from Docker or Azure
- Diagnostic trace logs for App
- Custom events and metrics

Getting started with Application Insights

Є декілька варіантів, як моніторити сервіси:

- At run time - підключити фреймворк до сервісу, який сам буде відправляти метрики
- At development time - можна збирати метрики під час розробки, щоб тестувати код
- Instrument your web pages - браузер буде відправляти метрики
- Analyze mobile app usage - потрібна інтеграція з Visual Studio App Center
- Availability tests - налаштувати ping до сервісів чи web сторінок

Discover log-based metrics

Є метрики, які зберігаються як окремі логи в таблицях. Є спеціальний синтаксис (Kusto Query Language), схожий на SQL, що дозволяє ці логи вичитувати. Далі на їх основі можна й графіки будувати, механізм має дуже багато можливостей.

Також є звичайні метрики, які попередньо агрегуються під час збору. Query Language більш швидко може їх обробити, але вони більш лімітовані. Такі метрики підходять для dashboards та нотифікацій.

Log-based metrics

Використовуючи SDK можна розробити власний механізм збору та відправки метрик до Azure. Але краще використовувати готові фреймворки, які автоматично це все обробляють. В такому варіанті більше ресурсів можна сфокусувати на налаштуванні візуалізації та діагностики в Azure Portal.

Деякі системи можуть генерувати забагато телеметрії. В такому випадку в App Insights розроблено можливості вибірки (sampling) та фільтрації. Це виправляє ситуація, але якість контролю/діагностики також падає.

Pre-aggregated metrics

Такі метрики не зберігаються як окремі логи/події. Це агреговані значення, які зберігаються з інтервалом в часі. Вони потребують набагато менше ресурсів та працюють швидше.

Щоб відрізнити типи метрик на UI в Azure Portal, вони так і підписані: Standard Metrics та Log-based metrics.

Останні версії SDK вміють агрегувати значення ще перед відправкою їх в Azure. Це дозволяє зменшити витрати.

Деякі SDK (наприклад, для браузера) не вміють цього робити. Тому в Azure розроблено механізм, який вміє агрегувати ці метрики з всіх подій (events), які надсилаються в Azure. Це працює трохи повільніше, але дозволяє мати майже real time. Також ingestion sampling не впливає на акуратність метрик.

Auto-instrumentation

На хості можна запустити окремий агент, який буде збирати та відправляти метрики. Це підтримується лімітованим списком сервісів Azure. Але цей список постійно росте.

<https://learn.microsoft.com/en-us/azure/azure-monitor/app/codeless-overview#supported-environments-languages-and-resource-providers>

Це підходить для сценаріїв, коли немає доступу до коду.

App Insights SDKs

Ці SDK потрібно встановлювати лише в наступних випадках:

- Потрібно збирати кастомні події чи метрики
- Потрібно контролювати flow телеметрії
- Механізм Auto-instrumentation не доступний

Програма може хоститись не тільки в Azure. SDK відправляє метрики використовуючи унікальний ID.

Деякі SDK підтримують distributed tracing з коробки. Наприклад, .NET, Java, Node.js, JavaScript.

Enable via OpenCensus

App Insights також підтримує distributed tracing через OpenCensus. Це open source бібліотека, що реалізує збір метрик. Також ця бібліотека корисна, коли потрібно збирати метрики з open source технологій, таких як Redis, Memcached, MongoDB та інших.

Availability Test

App Insights вміє надсилати web запити до сервісів.

В рамках одного ресурсу можна налаштувати до 100 тестів. Вони поділяються на:

- URL ping test (можна також налаштувати очікуваний час запиту, retries та інші функції)
- Standard test (preview) - це нова версія URL ping тесту, яка також включає перевірку SSL сертифікату, проактивну перевірку lifetime, кастомний контент (headers, body) в запиті
- Custom Track Availability test - можна створити власну програму для перевірки доступності, або сервіси самі можуть відправляти статус як частину метрик (TrackAvailability метод з SDK)

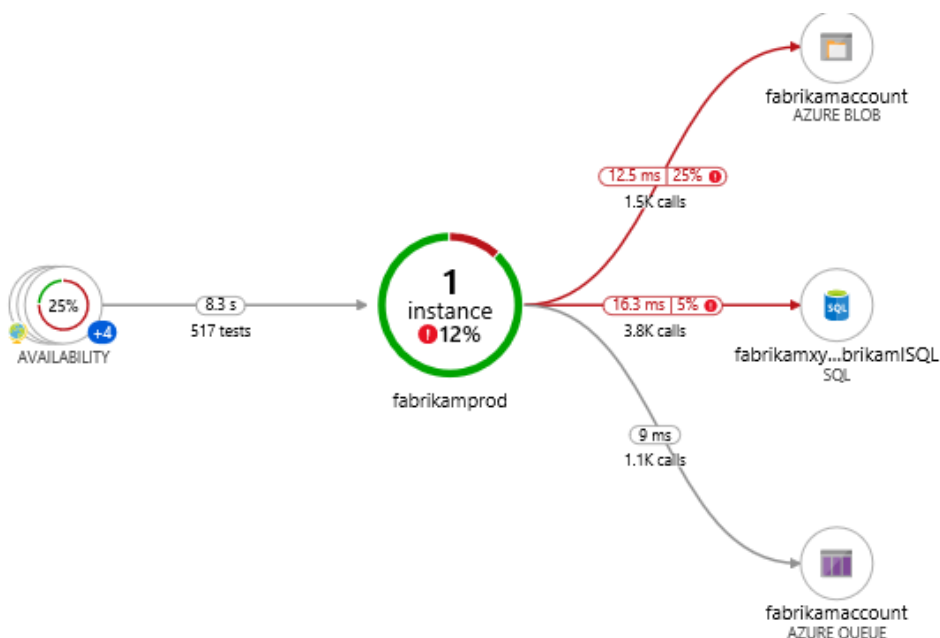
URL ping використовує публічні DNS сервера, щоб знаходити пару domain name - IP Address. Якщо в проєкті використано приватні DNS сервера, то цей механізм не буде працювати.

Application Map

Це діаграма, яка включає всі сервіси, інстанси та ресурси, які є в системі. Також вона відображає HTTP трафік, час запитів, помилки.

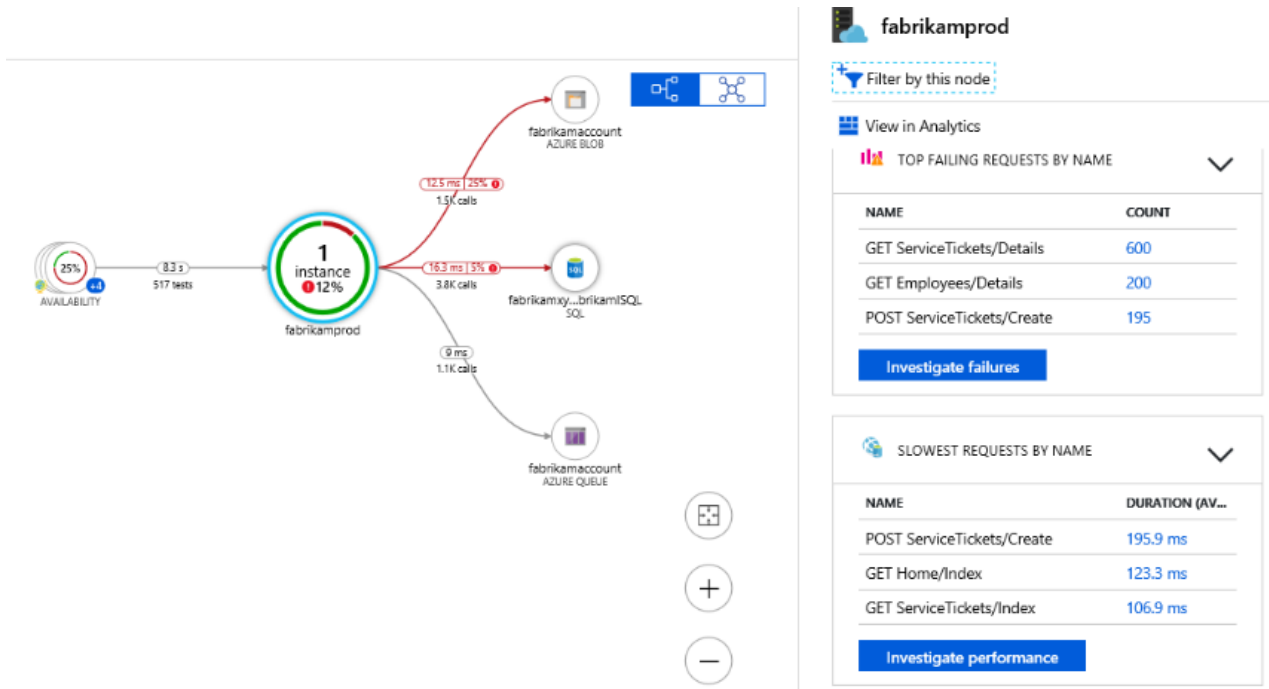
Діаграма інтерактивна, можна взаємодіяти з кожним елементом для відкриття більш детальної інформації.

Application Map орієнтується на HTTP трафік. Під капотом механізм запускає ті ж самі Kusto Query, щоб отримати метрики.



На цьому зображенні приклад Application Map для одного мікросервісу, який використовує Blob Storage, SQL Server та Storage Queue. Також є декілька availability tests. По трафіку видно, що деякі HTTP запити мають від 5 до 25% failure rate.

Якщо натиснути на конкретний елемент, то відкриється додаткова інформація про нього.



Щоб відокремити типи або групи мікросервісів, потрібно використовувати cloud role name. Це щось типу унікального ідентифікатора або display name.