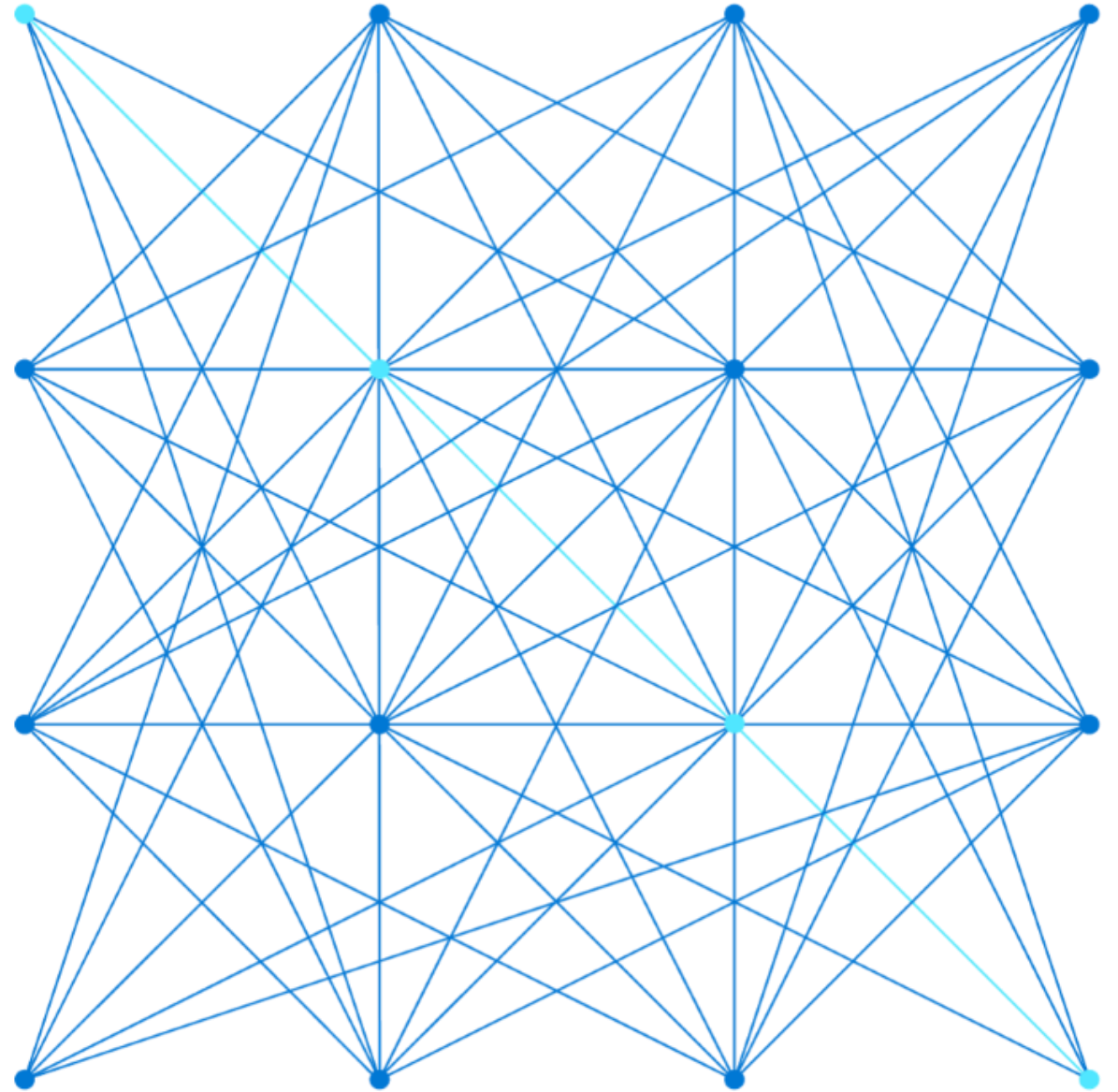


AZ-400.00

Module 17: Implementing Feedback for Development Teams



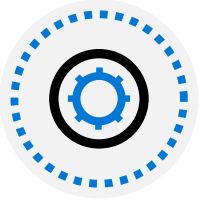
Lesson 01: Module overview



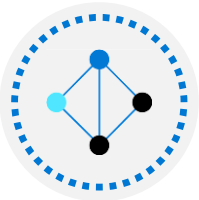
Module overview



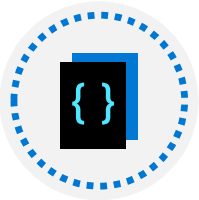
Lesson 1: Module overview



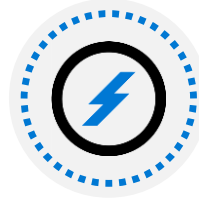
Lesson 2: Implement tools to track system usage, feature usage, and flow



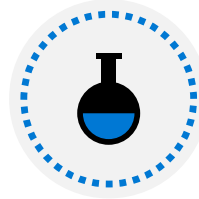
Lesson 3: Implement routing for mobile application crash report data



Lesson 4: Develop monitoring and status dashboards



Lesson 5: Integrate and configure ticketing systems



Lesson 6: Lab



Lesson 7: Module Review and Takeaways

Learning objectives

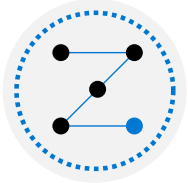
After completing this module, students will be able to:



Configure crash report integration for client applications



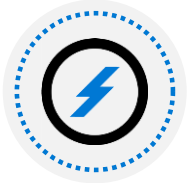
Develop monitoring and status dashboards



Implement routing for client application crash report data

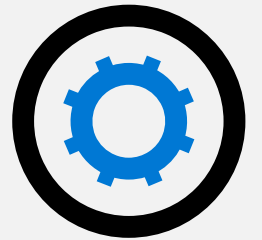


Implement tools to track system usage, feature usage, and flow

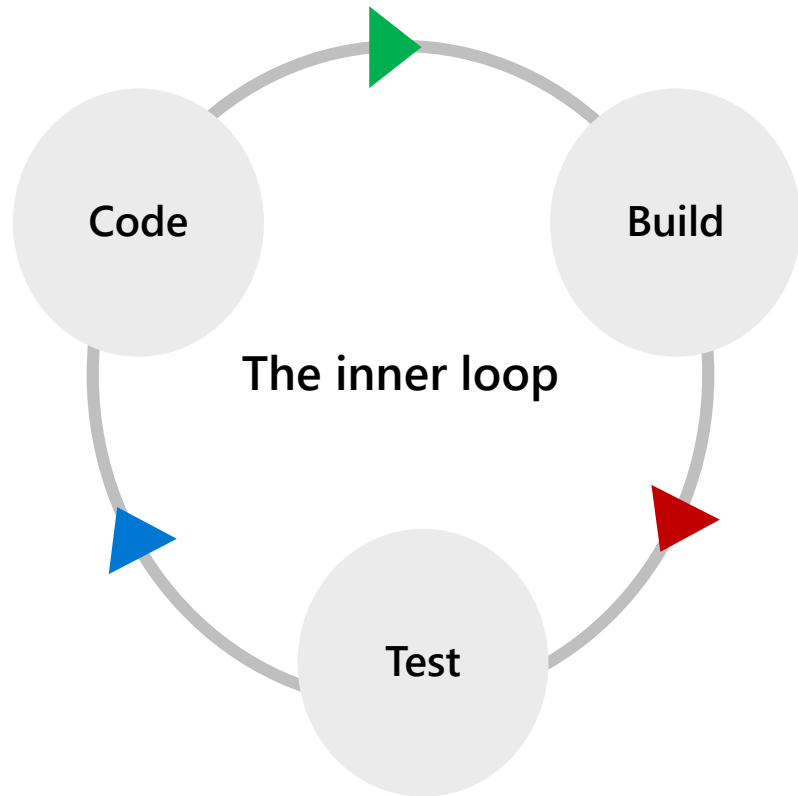


Integrate and configure ticketing systems with development team's work management

Lesson 02: Implement tools to track system usage, feature usage, and flow



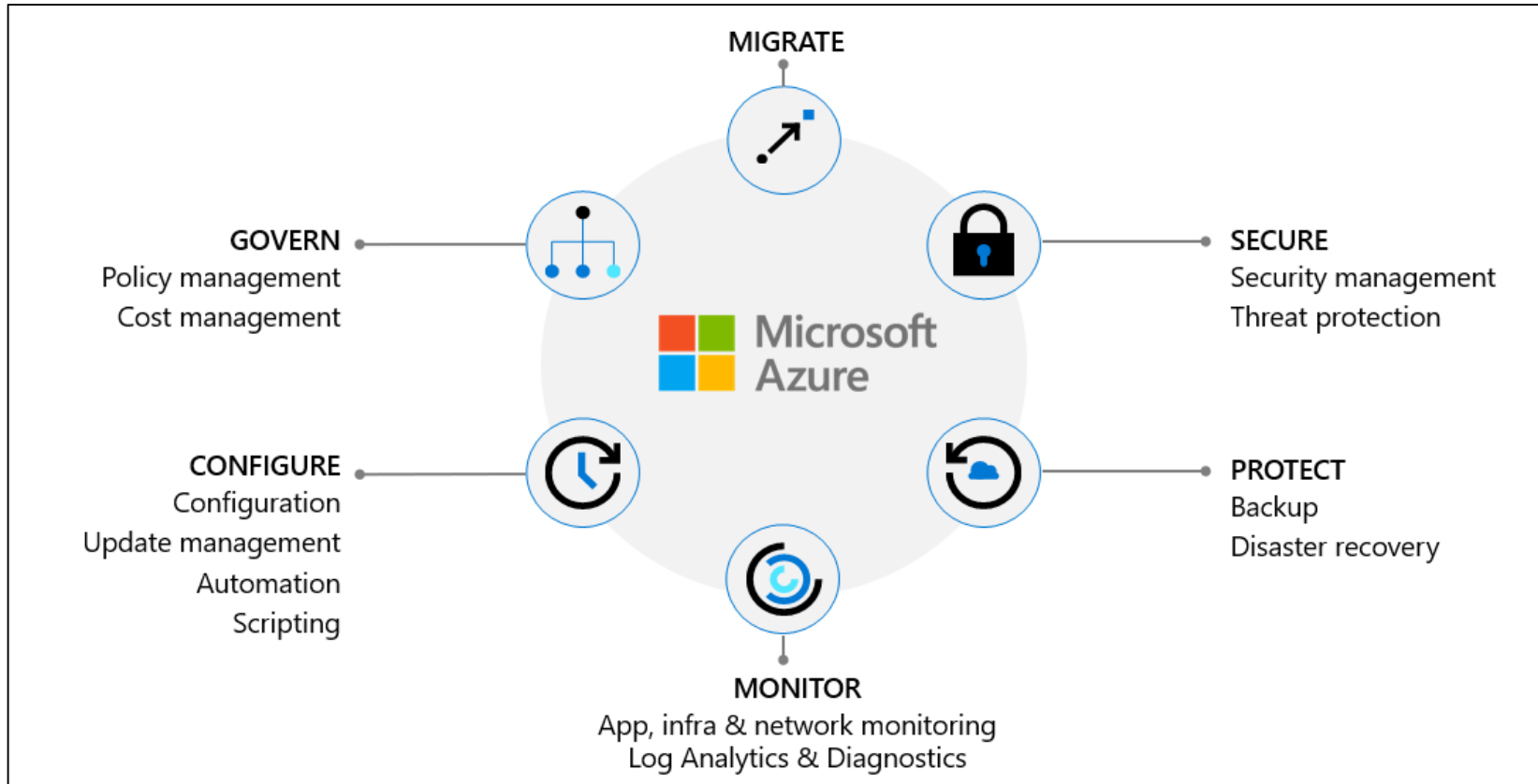
The inner loop



“the iterative process that a developer performs when they write, build, and debug code”

- An important goal is to speed up the cadence of the inner-loop.
- Doing so means getting new work into production asap.
- There are several things that a team can do to optimize the inner loop **for larger codebases**:
 - Only build and test what was changed.
 - Cache intermediate build results to speed up full builds.
 - Break up the codebase into small units and share binaries.

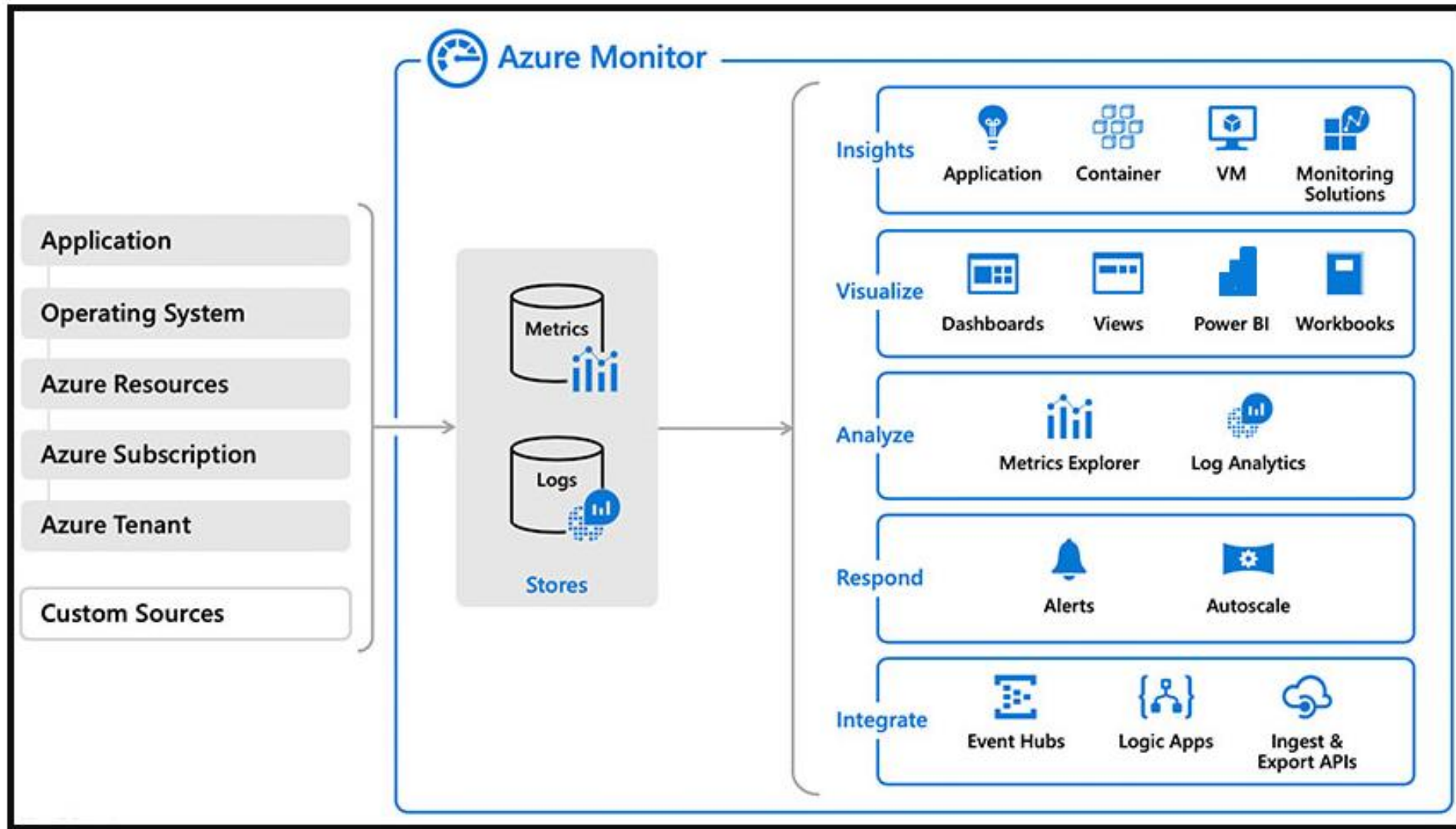
Introduction to continuous monitoring



“Continuous monitoring refers to the process and technology required to incorporate monitoring across each phase of your DevOps and IT operations lifecycles”.

#mw- careful. This diagram is only concerned with *operating* an Azure environment. See notes for more links on other monitoring

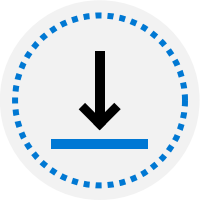
Azure Log Analytics



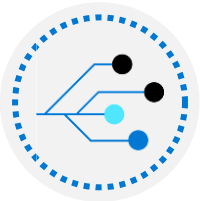
Kusto Query Language (KQL)



KQL is the language used to query Log Analytics

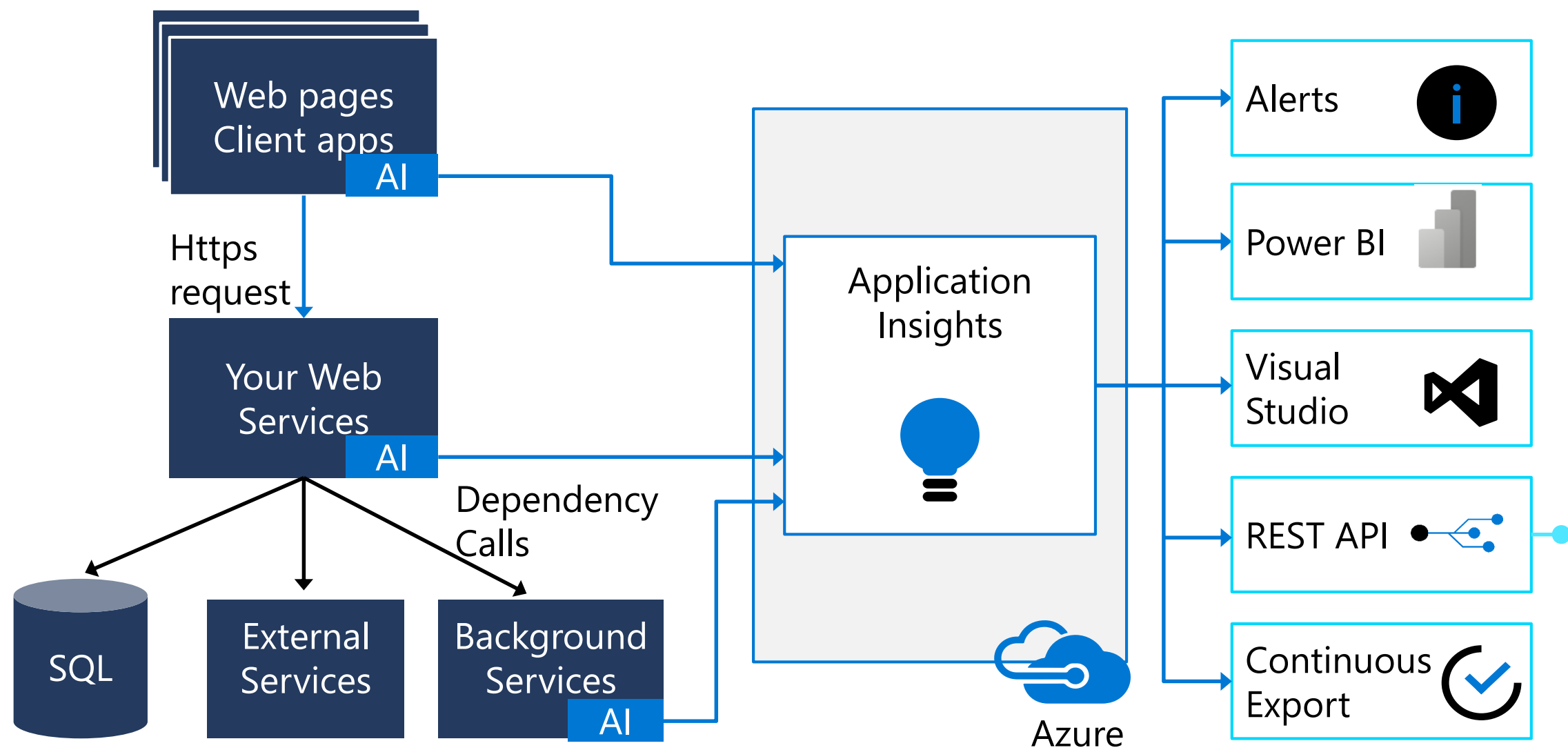


Supports queries and control commands



Supported within Azure Data Explorer and Azure Data Studio (with notebook support)

Application Insights

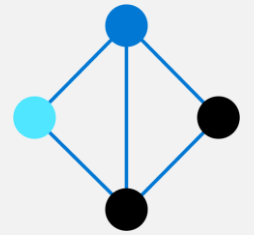


Demonstration: Adding Application Insights to an ASP.NET core application

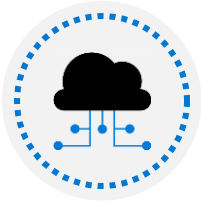
DEMO

#mw-d-ApplicationInsights

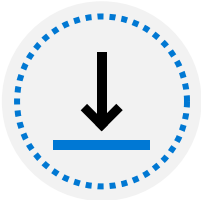
Lesson 03: Implement routing for mobile application crash report data



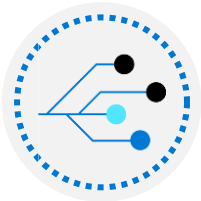
App Center Diagnostics



[App Center](#) Diagnostics is a cloud service that helps developers monitor the health of an application, delivering the data needed to understand what happens when an app fails during testing or in the wild



In the [App Center Diagnostics UI](#), you can attach, [view and download one binary and one test attachment to your crash reports](#)



Track events leading up to a crash to capture useful information about the state of your app

Configure alerts

1

Log into App Center and select your app

2

In the left menu, navigate to **Settings**

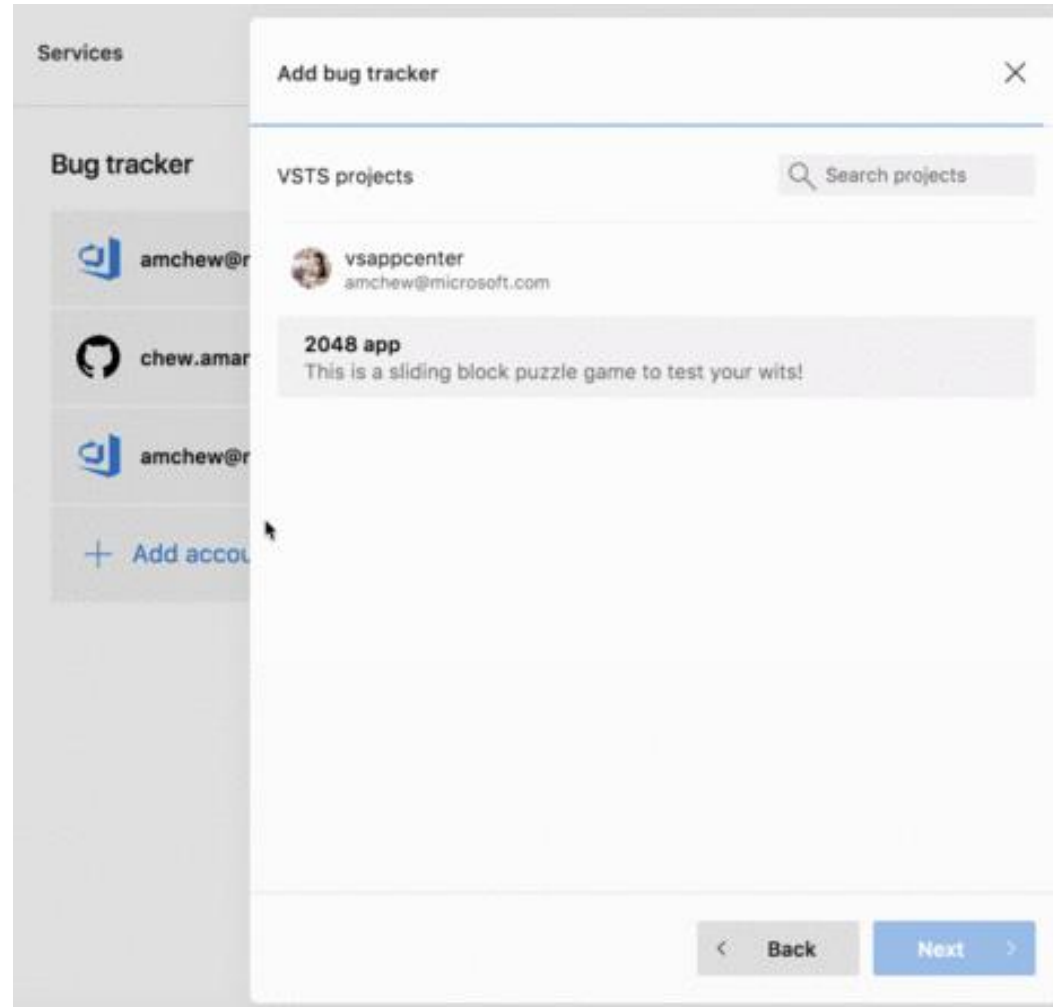
3

Click on **Email Notifications**

4

Select when a new crash group is created

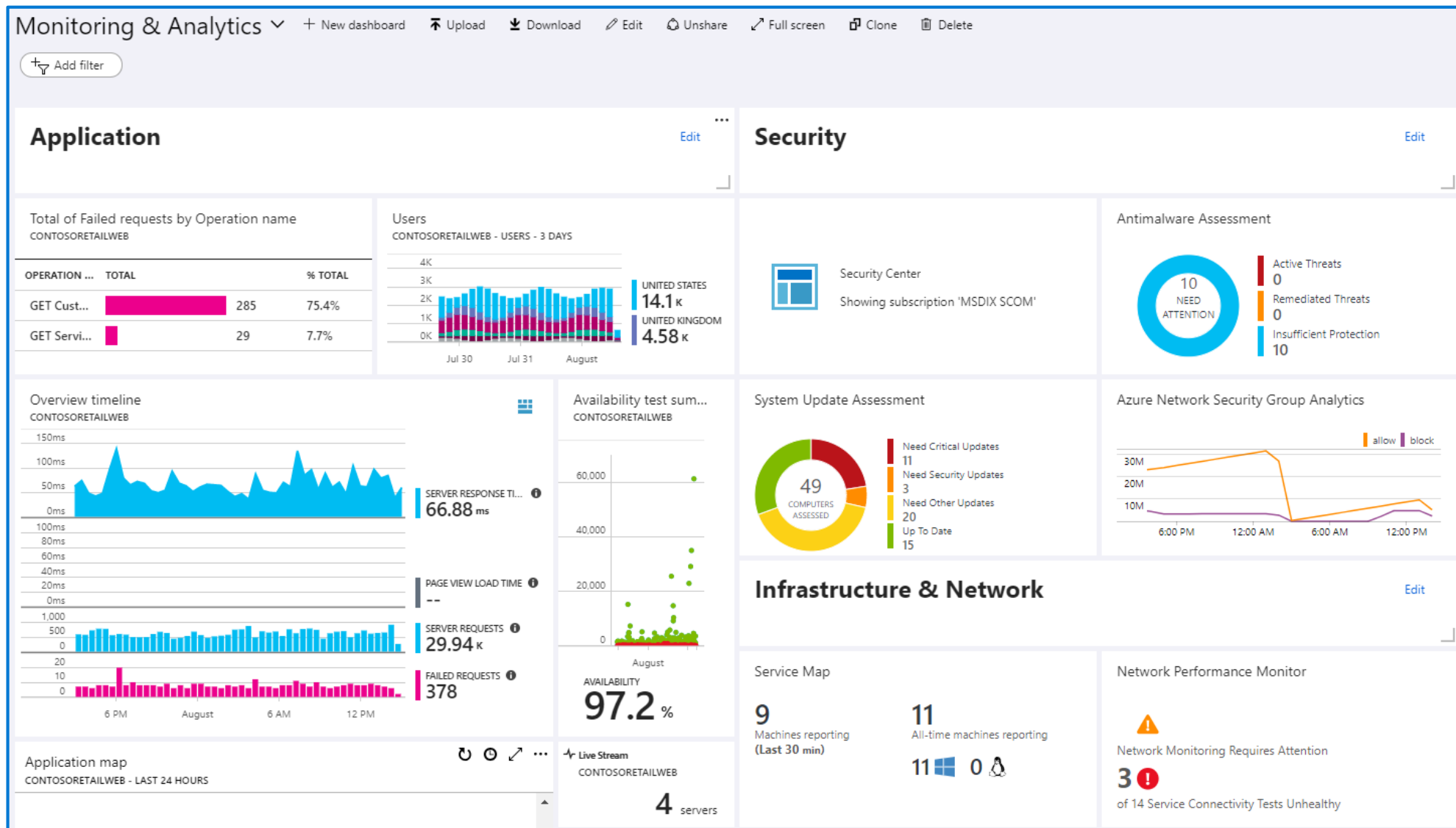
Create a bug tracker



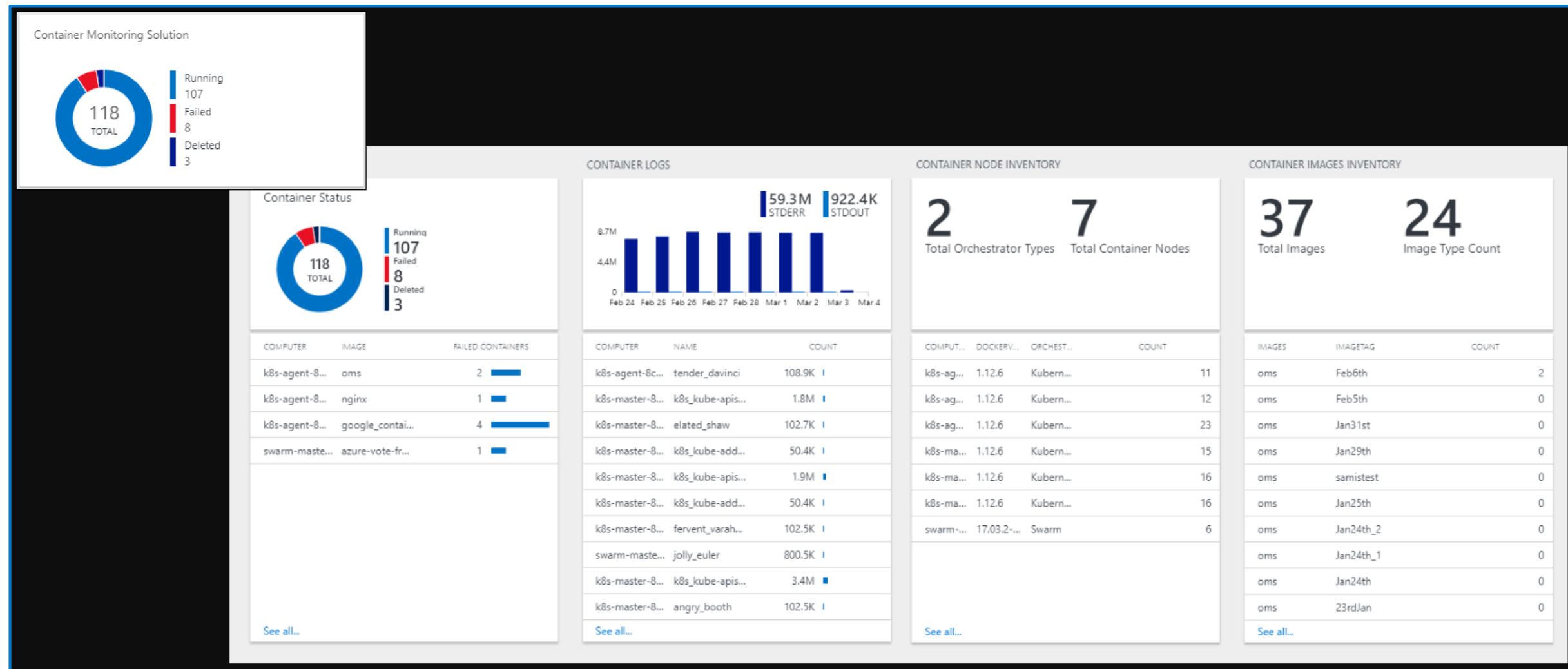
Lesson 04: Develop monitoring and status dashboards



Azure Dashboards



View Designer in Azure Monitor



Azure Monitor workbooks

Analysis of Page Views

Page views correspond to user activity in your app. Understanding how your users interact with your pages will give you good insights into what is working in your app and what aspects need improvements.

This report will help

- Usage
- Time spent on p
- Time to first int
- Exit rates

If your telemetry do

Pages: Home Page, D

OptimizeCalculations

Usage

This section helps you understand how

Page N...	Unique Users	As % of
Overall	8813	
Home P...	8811	
Create	8792	
Details	8780	

- The As % of App Users/Sessions/V

Time Spent on Page

This report helps you understand the time spent by customers in your pages. Longer time spent pages usually indicates good engagement and is generally the desired behavior.

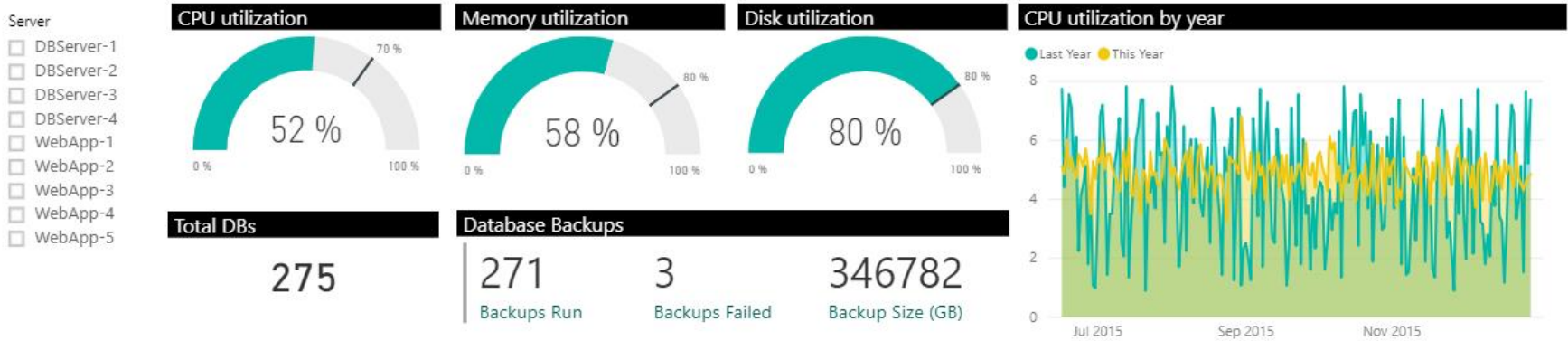
Ignore Durations Over: 3600s

Page Name	Sampled Page ...	Median (second...	75th Percentile (seconds)	90th Percentile ...	Mean (seconds)
Overall	107150	149.7	375	750.5	294.3
Create	29522	174.4	540	896.5	348.1
Details	33086	169.2	480	900	351.5
Home Page	44542	105.6	300	500	216.2

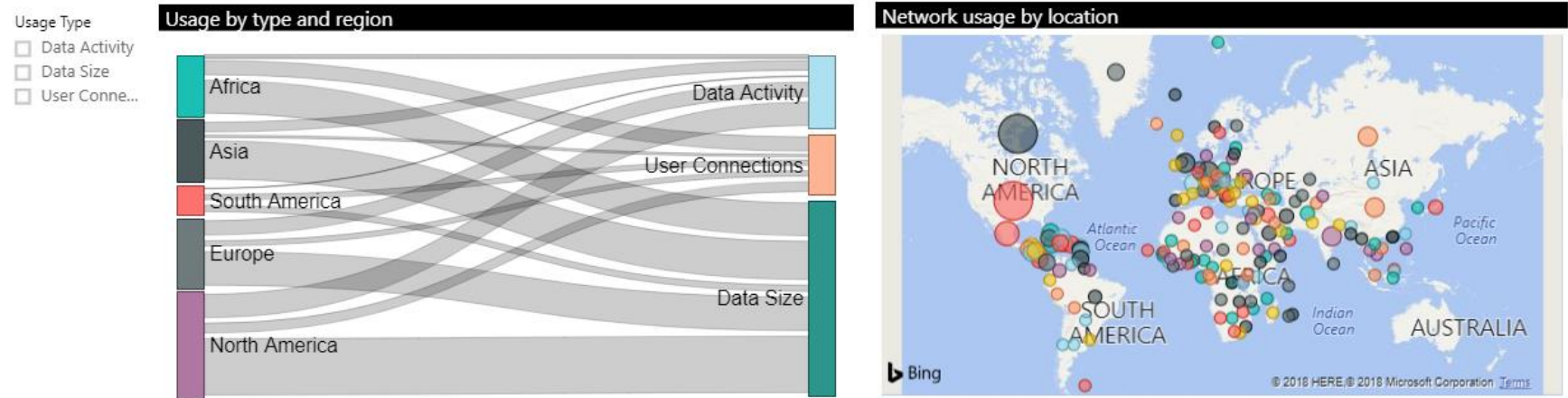
- The calculations may use sampling based on the `optimizeCalculationsFor` parameter.
- Time Spent on Page does not consider exit pages (last page of the session) in this calculations. The `Sampled Page Views` column may be fewer than the sampling count of 100000 because of this.

Power BI

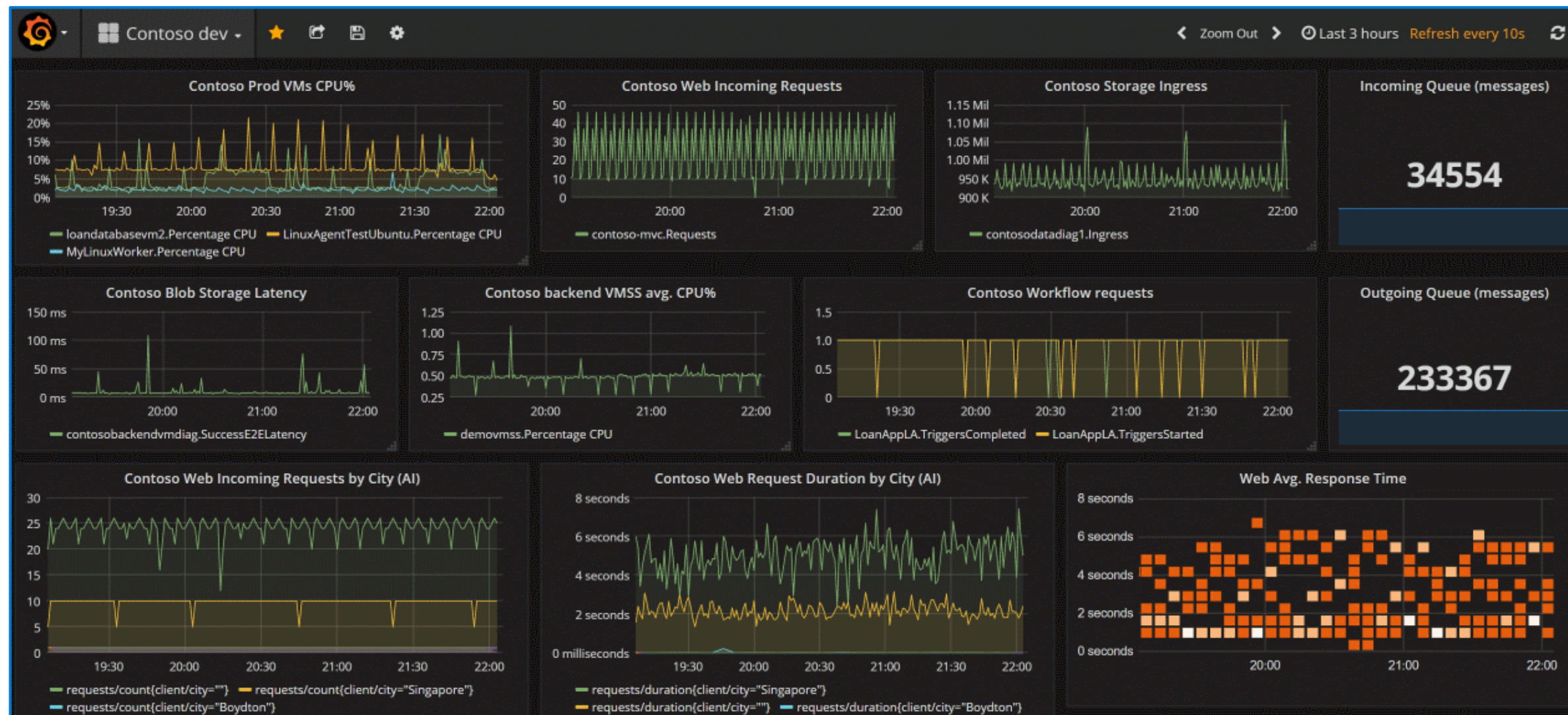
IT Operations



Resource usage and availability



Grafana



Build your own custom application

Advantages:

Complete flexibility

Combine metrics and log data

Disadvantages:

Significant engineering required

Lesson 05: Integrate and configure ticketing systems



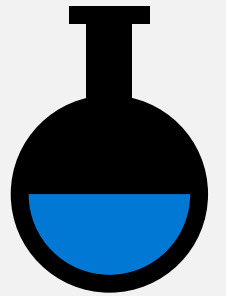
IT Service Management Connector

With ITSMC, you can:

1. Create work items in your ITSM tool, based on your Azure alerts (Metric Alerts, Activity Log Alerts, and Log Analytics alerts).
2. Optionally, you can sync your incident and change request data from your ITSM tool to an Azure Log Analytics workspace.

(#mw – See: <https://docs.microsoft.com/en-us/azure/azure-monitor/alerts/itsmc-overview>)

Lesson 06: Lab



Monitoring application performance with Application Insights



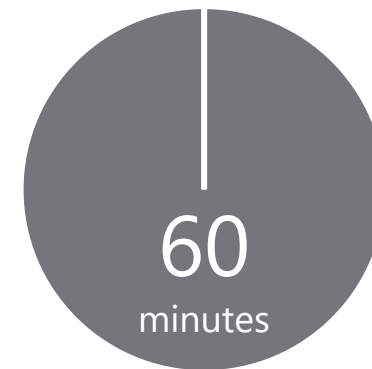
Lab overview:

In this lab, you'll learn about how you can add Application Insights to an existing web application, as well as how to monitor the application via the Azure portal.

Objectives:

- Deploy Azure App Service web apps
- Generate and monitor Azure web app application traffic by using Application Insights
- Investigate Azure web app performance by using Application Insights
- Track Azure web app usage by using Application Insights
- Create Azure web app alerts by using Application Insights

Duration:



Lesson 07: Module review and takeaways



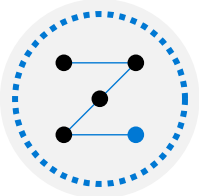
What did you learn?



Configure crash report integration for client applications



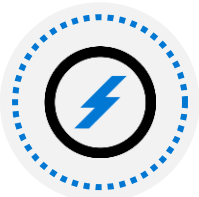
Develop monitoring and status dashboards



Implement routing for client application crash report data



Implement tools to track system usage, feature usage, and flow



Integrate and configure ticketing systems with development team's work management

Module review questions

1 Does Azure Monitor allow you to create alerts from log queries?

2 What features are provided by Azure Monitor?

3 What query language can you use to query Azure Log Analytics?

4 What platform integrations does Azure Monitor provide to visualize your logs in real time?