



Practical Exercises: Application Monitoring and Feedback Loops

Prerequisites

- Visual Studio 2017 Enterprise.
- Download the [Parts Unlimited project from GitHub](#).
- Azure Subscription.
- VSTS project.



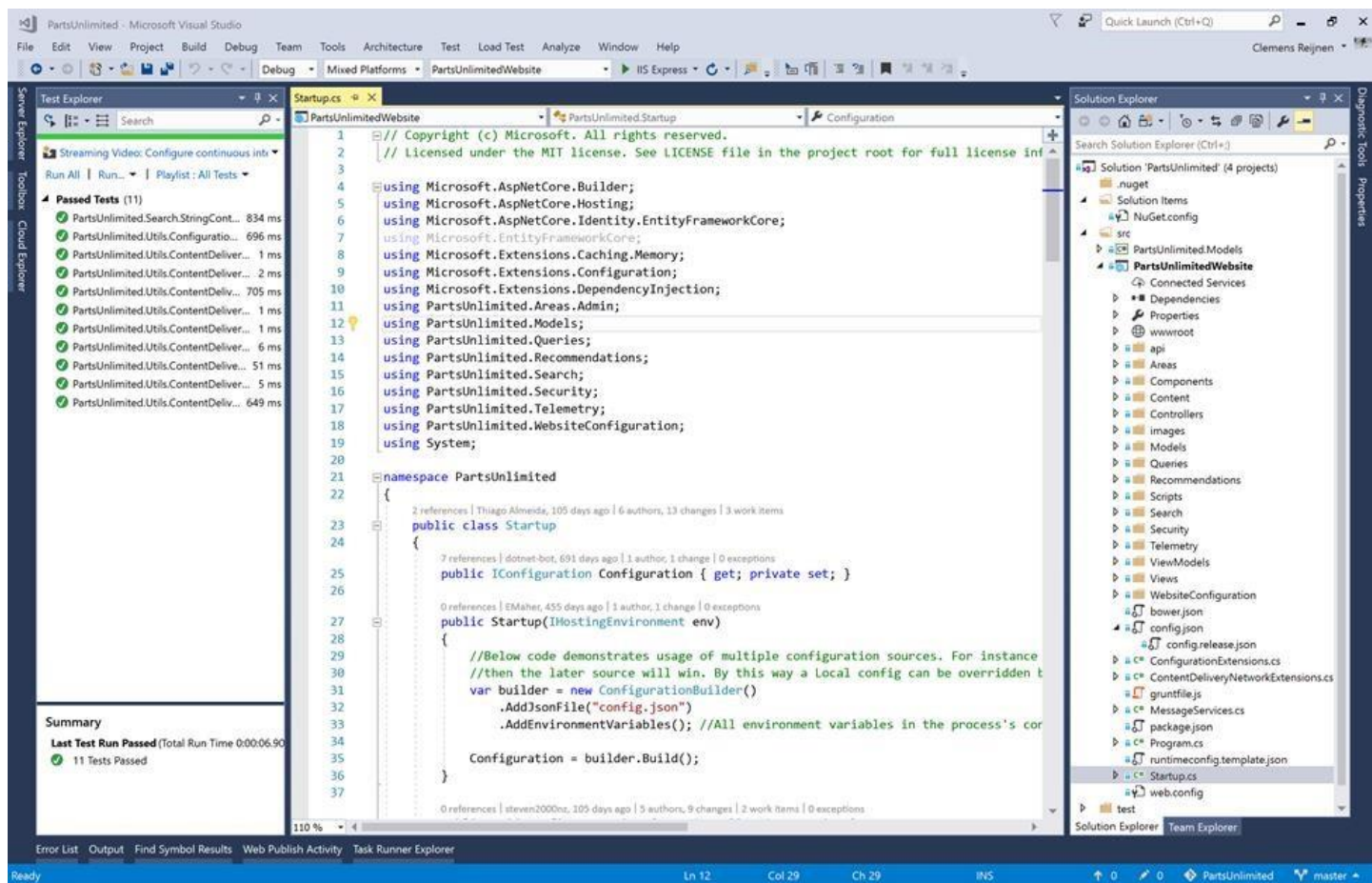
Scenario 1: APM with Application Insights

To maximize the availability and time-to-recover of the PartsUnlimited website, the operations team wants to track all requests performed on the website. The good and failing requests for the server and depended calls should be reported.

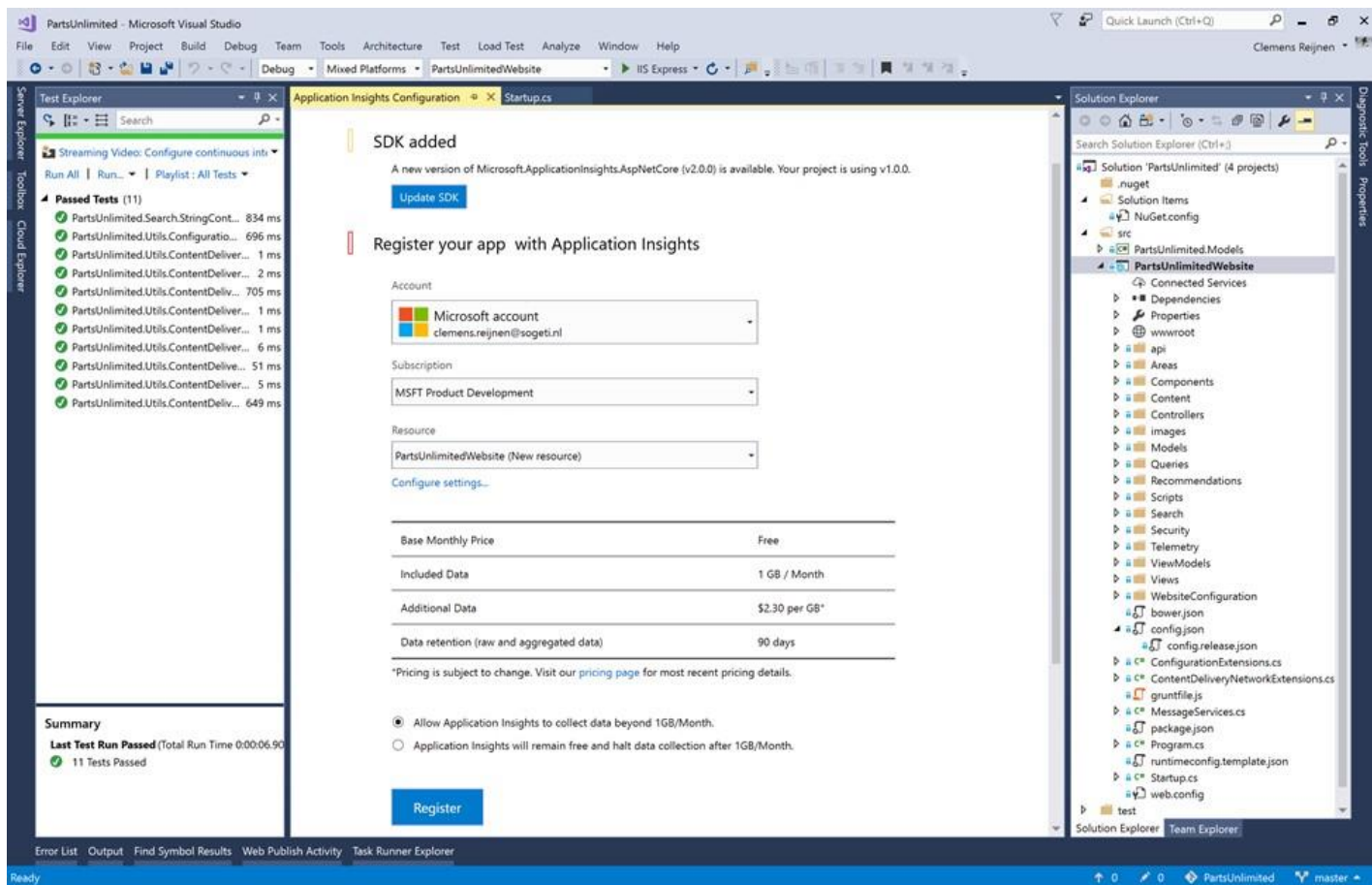
Implement and configure, as a developer on the PartsUnlimited DevOps team, the monitoring and reporting of telemetry data.

Lab 1: Add Application Insights Telemetry to the Parts Unlimited Website

1. Open the PartsUnlimited solution within Visual Studio. Notes:
 - VS2017 may need to migrate the project files.
 - Note: npm and grunt must be installed.
 - **Note:** set the *config.json* and *appsettings.json* properties to **Copy if newer** instead.
- a) Restore Packages
- b) Compile the solution
- c) Make sure the unit tests execute successfully

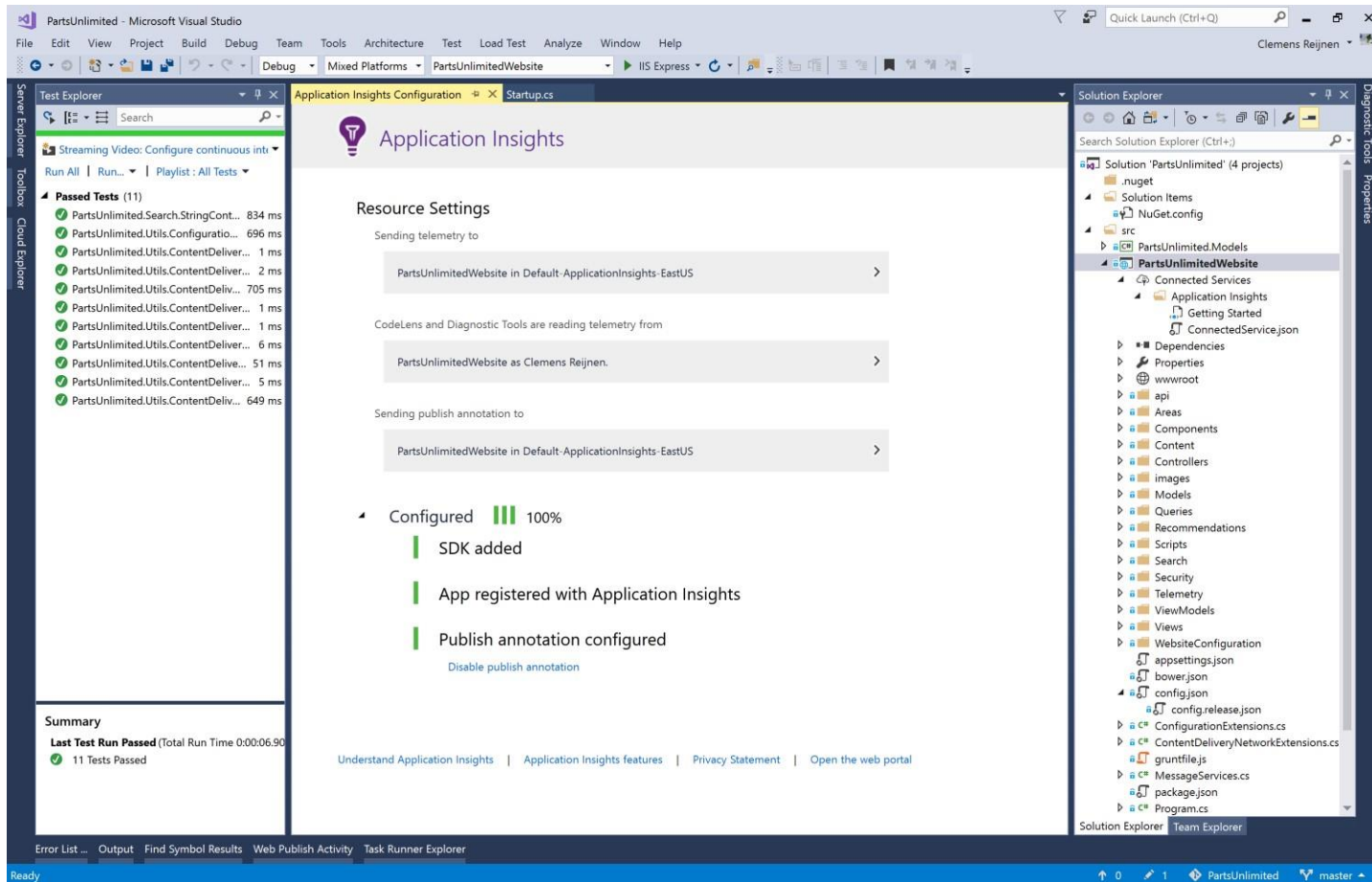


2. Add Application Insights to the PartsUnlimited Website.
 - a) Select the PartsUnlimited Website, right -click, add Application Insights.
 - b) Create new Application Insights resource. Update the SDK and select your Azure subscription.



c) Configure resource settings:

- i. Telemetry
- ii. CodeLens and Diagnostics



3. Add and check the Application Insights resources (see also, <https://github.com/Microsoft/ApplicationInsights-aspnetcore/wiki/Getting-Started>).

- Install NuGet package Microsoft.ApplicationInsights.AspNetCore
- Add appsettings.json instrumentation key
- Add to startup.cs to the Application Insights instrumentation code

i. In the startup method: appsettings.json and the isDevelopment statement.

```
var builder = new ConfigurationBuilder()
    .AddJsonFile("config.json")
    .AddJsonFile("appsettings.json")
    .AddEnvironmentVariables(); //All environment variables in the

if (env.IsDevelopment())
{
    builder.AddApplicationInsightsSettings(developerMode: true);
}

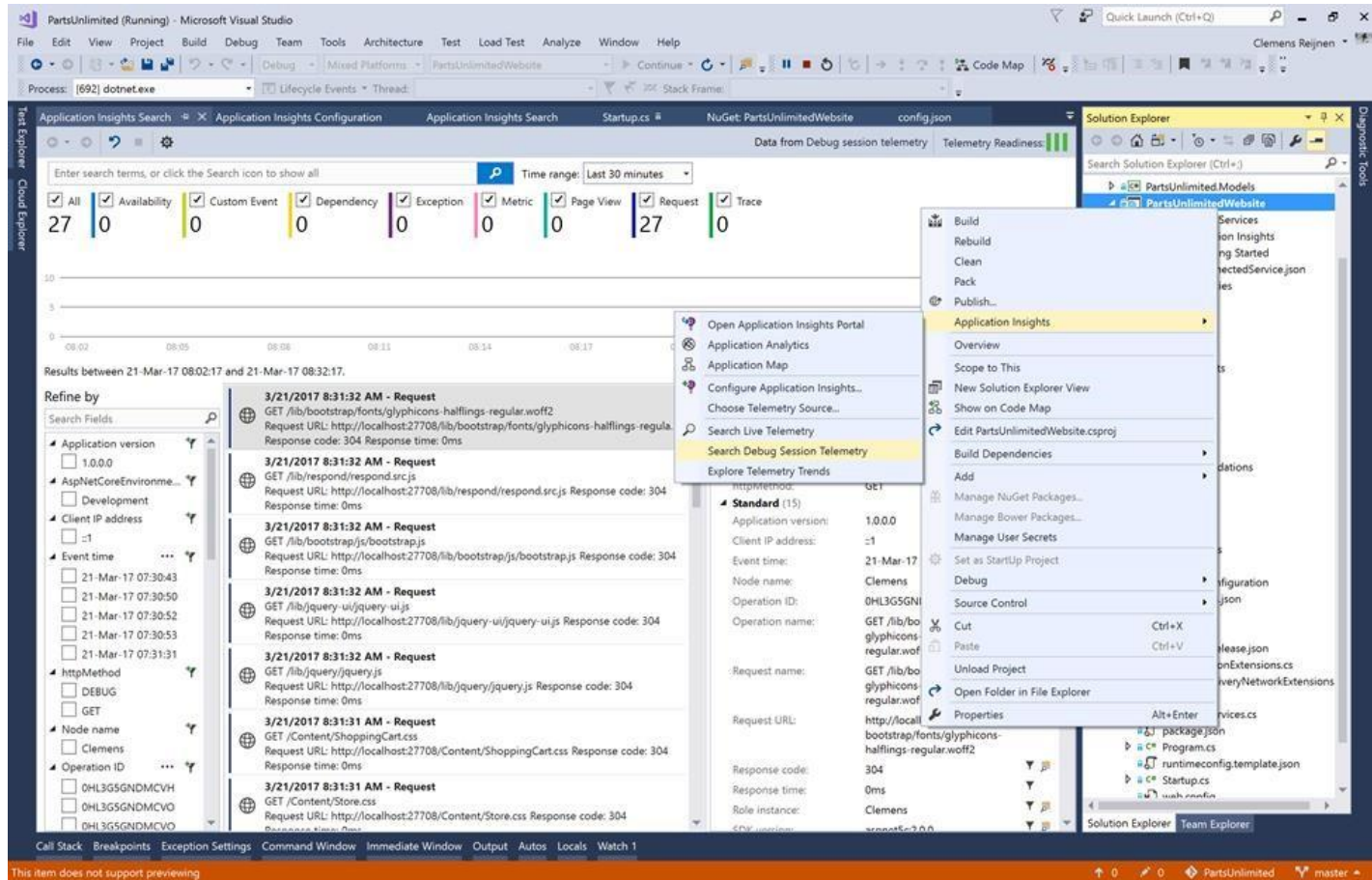
Configuration = builder.Build();
```

ii. In the ConfigureServices method:

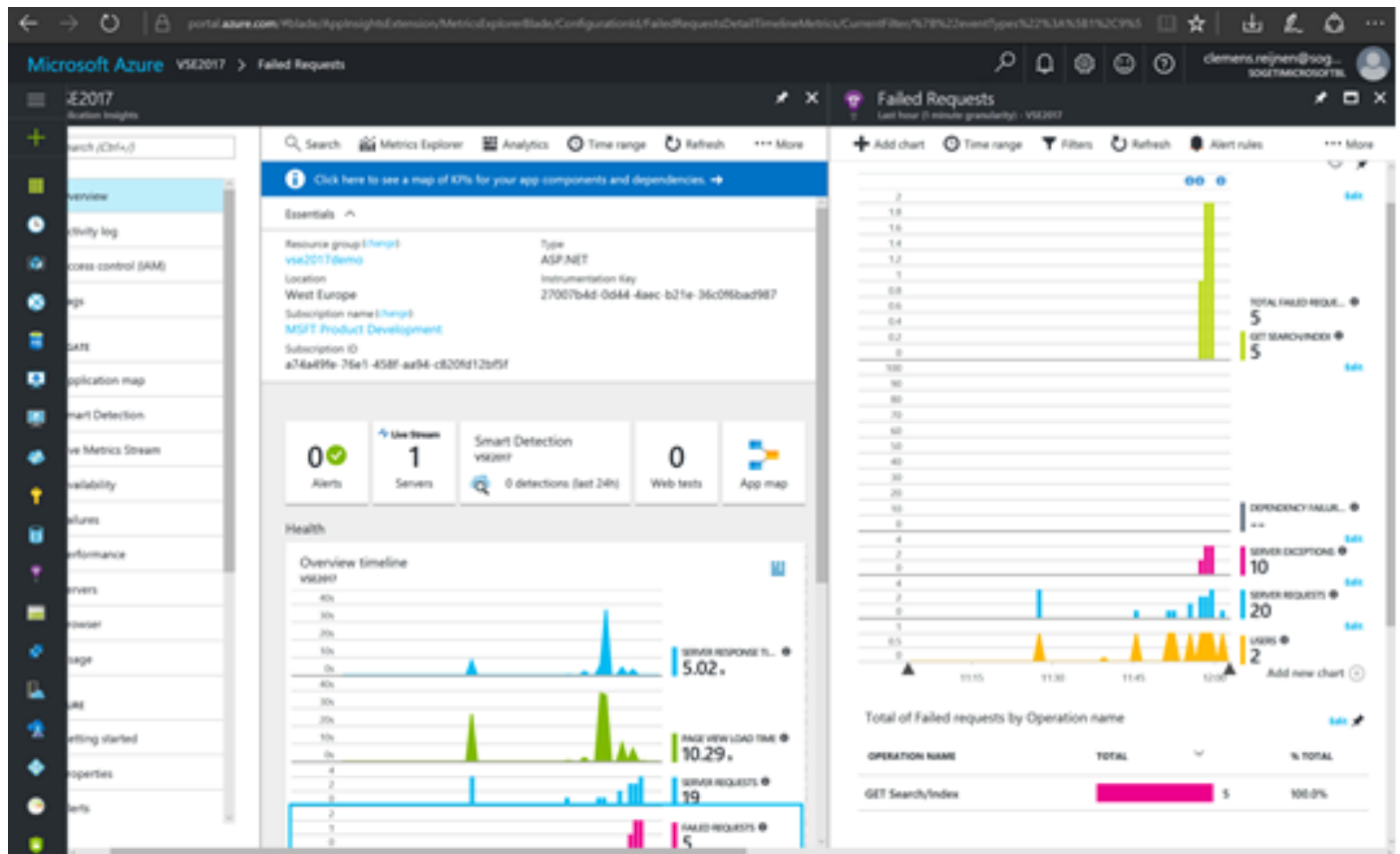
```
services.AddApplicationInsightsTelemetry(Configuration);
```

4. Create telemetry data by running the website.

- a) Start the application from Visual Studio. Press F5.
 - b) Browse website and search for products.
5. See your telemetry in Visual Studio.
- a) In Visual Studio, open the Application Insights window. Either click the Application Insights button, or right-click your project in Solution Explorer, select Application Insights, and then click Search Debug Telemetry.



6. Investigate telemetry data.
- a) Open Azure Portal, goto portal.azure.com
 - b) Navigate to the just created Application Insight resource.
 - c) Investigate reports.



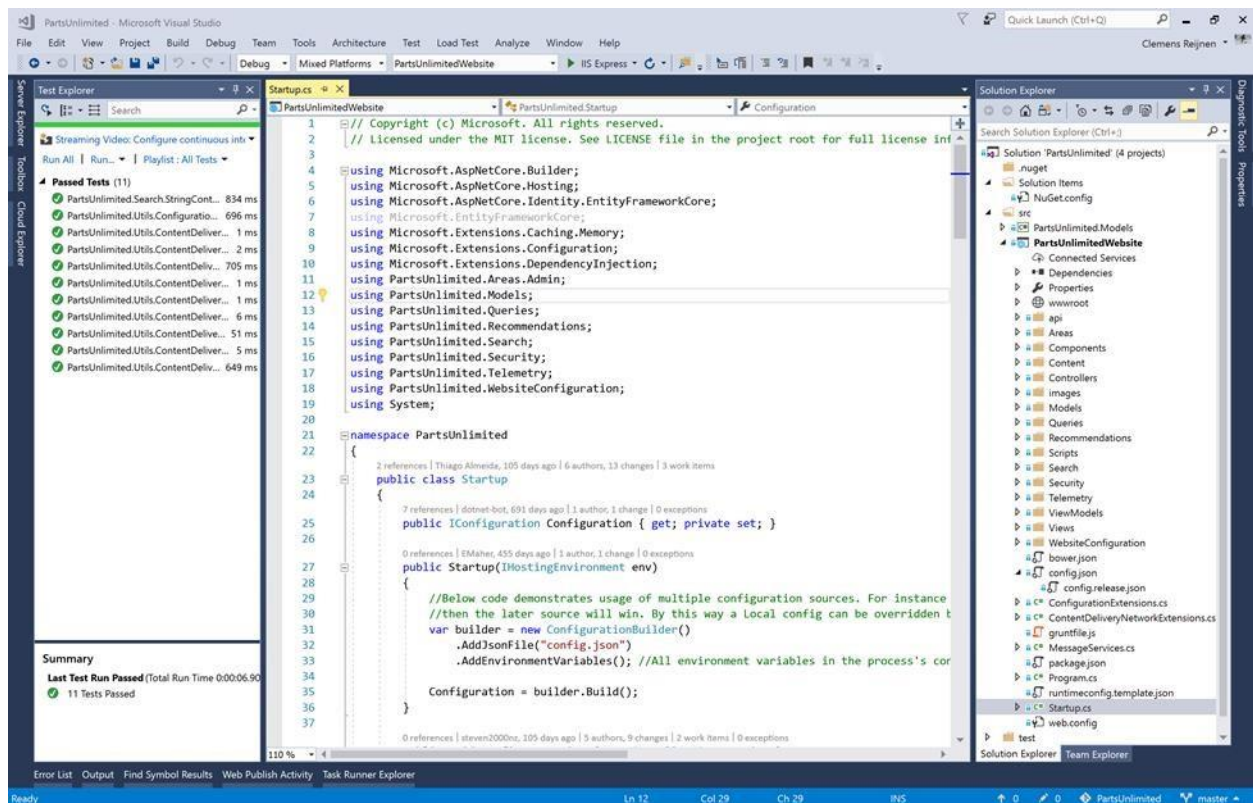
Scenario 2: User Telemetry

As the business owner of PartsUnlimited, you want to track how the Sales site is used so that you know which part of the site tracks the most users and by whom. When a new feature is added or changed, you want to see the impact of the change with a mark on the telemetry data.

Implement and configure, as a developer, the usages data of the website. Show the telemetry data in the Azure portal and separate it by release. Make sure the different stages report to different reports.

Lab 2: Add Usages Telemetry Metering

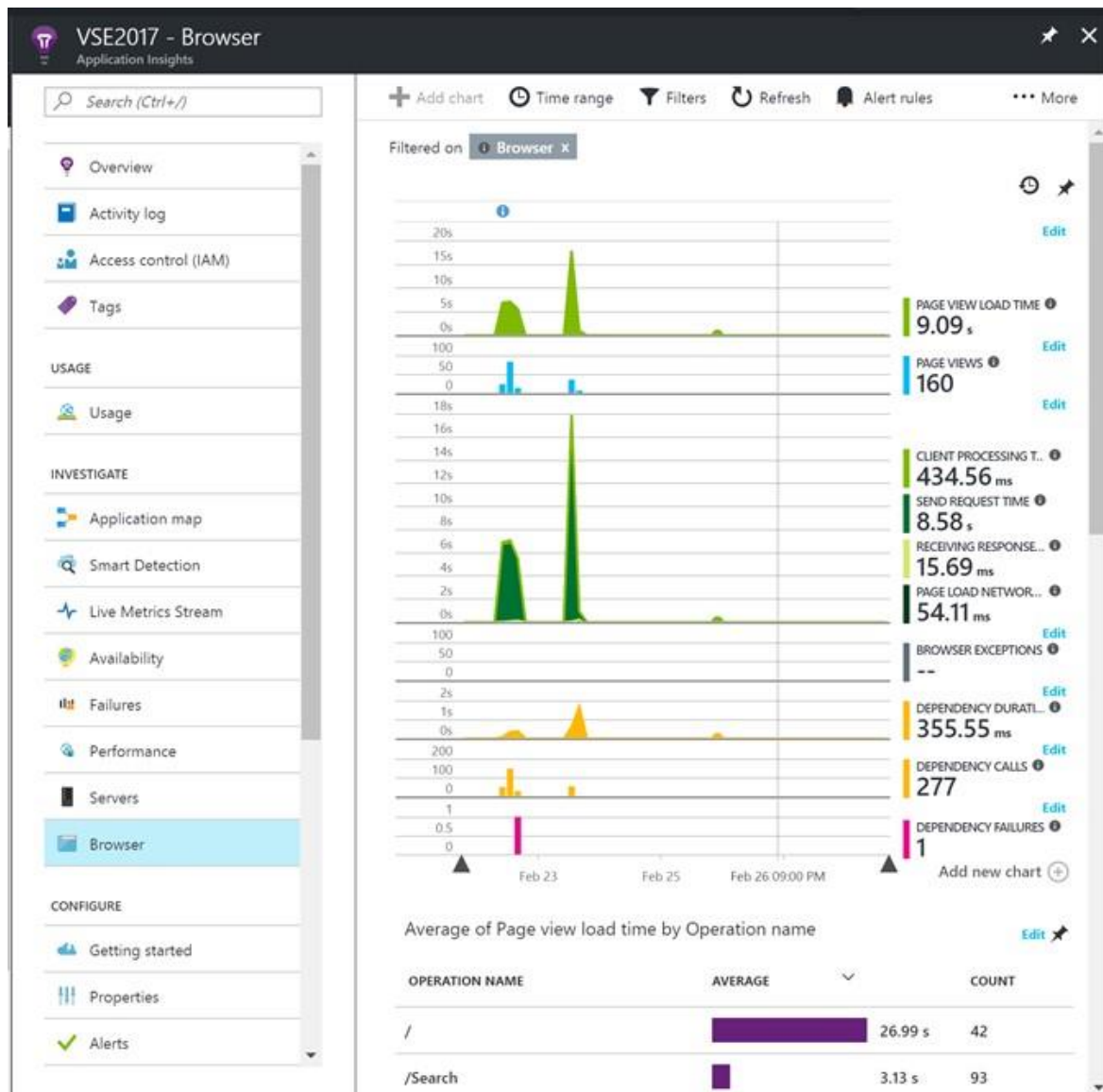
- Open PartsUnlimited within Visual Studio from Lab1.
 - Restore Packages
 - Compile the solution
 - make sure the unit tests execute successfully



2. Add or validate the web front-end monitoring for user telemetry.
 - a) Go to the `_Layout.cshtml` file.
 - b) Add `@inject Microsoft.ApplicationInsights.AspNetCore.JavaScriptSnippet` `JavaScriptSnippet`.
 - c) Add before the closing head section `@Html.Raw(JavaScriptSnippet.FullScript)`

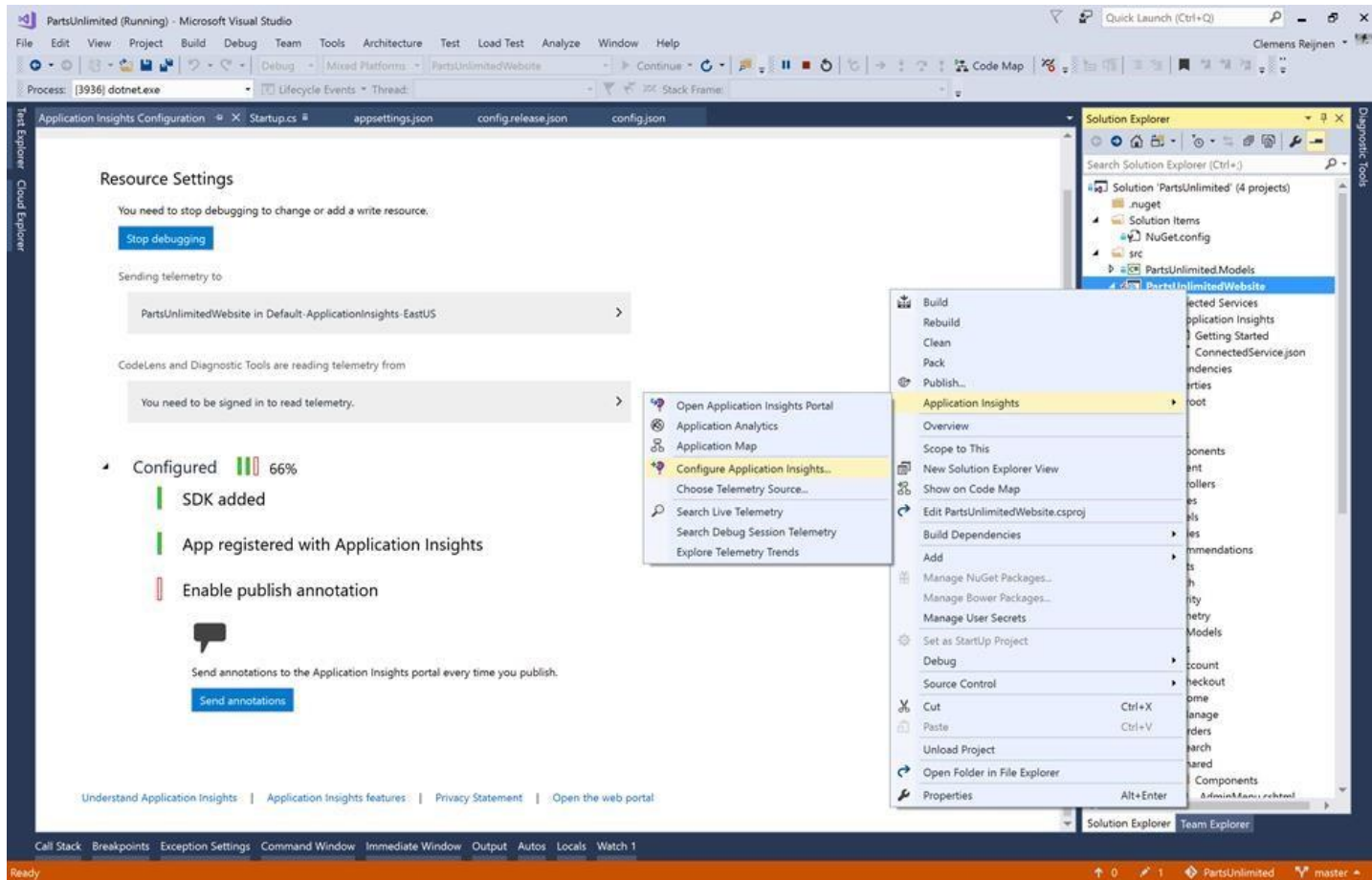
Lab 3: Make Telemetry Metering Stage Specific

1. Make the instrumentationKey dynamic so it can be set for different stages.
 - a) Add an instrumentation key section to config to json and to the config.release.json.
 - b) Remove the `.AddJsonFile("appsettings.json")` from the startup.cs.
2. Create telemetry data by running the website and browse the products.
3. Investigate created front-end request and response data.

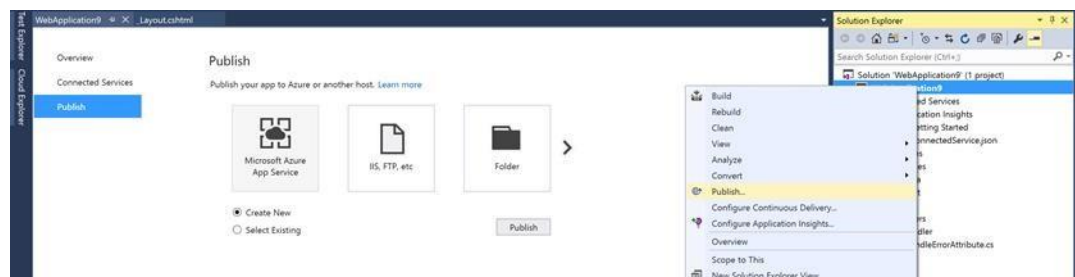


Lab 4: Set Version Publish Annotations to Telemetry Graphs

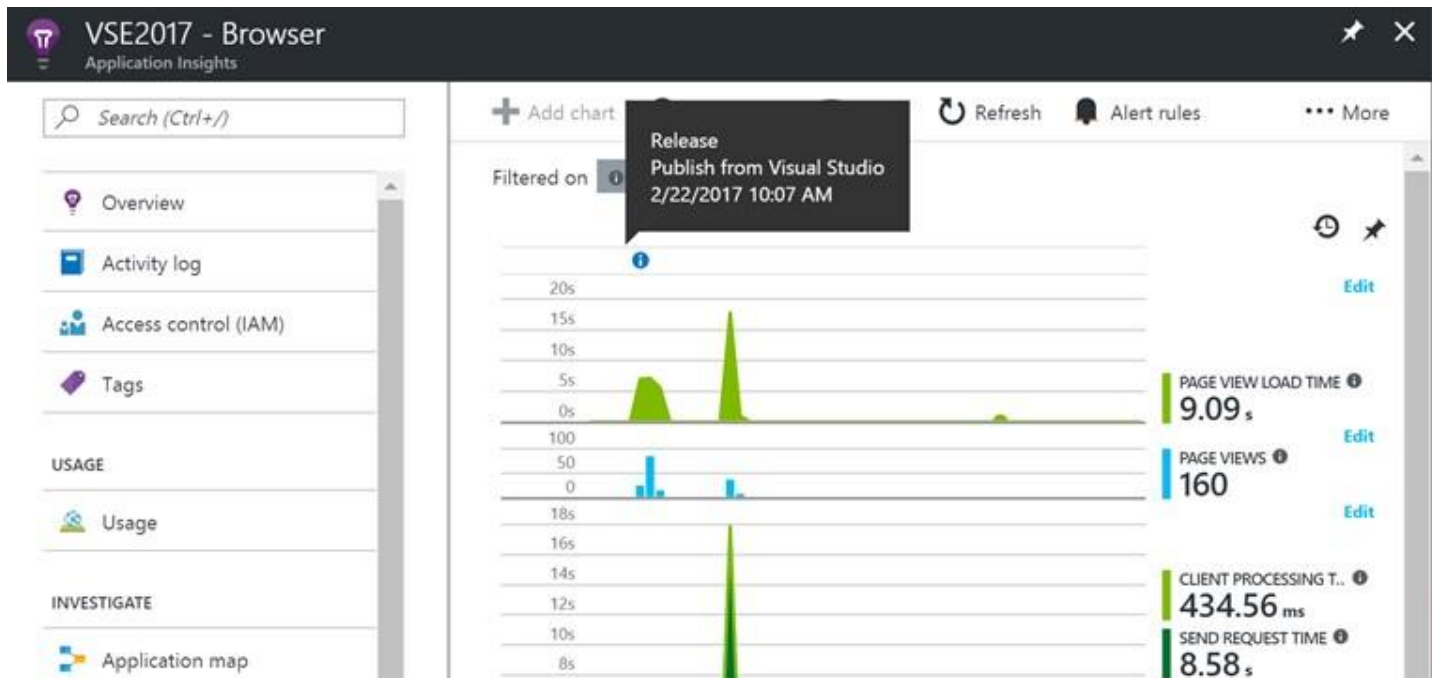
1. Add publish annotations Application Insights configuration.
 - a) Stop debugging.
 - b) Open the Application Insights Resource Settings.
 - c) Configure Publish annotations if not done already.



2. Publish the PartsUnlimited website to Azure WebApps and see the publish annotation in the Application Insights graphs.
 - a) Right-click on the PartsUnlimited WebApp and select Publish WebApp.



- b) Create Azure Resource "PartsUnlimited-RG".
 - c) Publish PartsUnlimited website.
 - d) Browse the PartsUnlimited website.
 - e) Open the Azure portal Application insight resources see the publish annotation



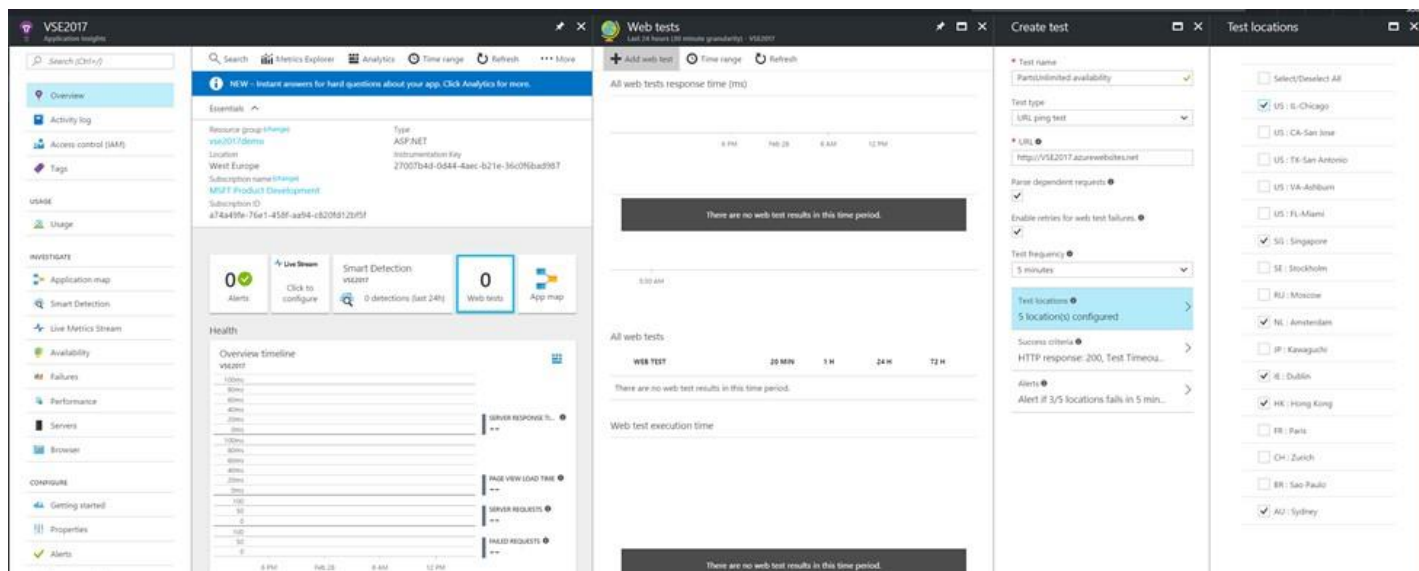
Scenario 3: Health and Performance Monitoring

The sales of PartsUnlimited is expanded globally to multiple regions. These regions are Asia, West Europe, and North Europe. From Asia, a lot of traffic is expected on the search part of the website, due to interest in Jumper Leads from that area. The product owners from PartsUnlimited want to monitor the availability of the website from all regions. Next to availability monitoring, they also want to be sure the website stays responsive when an enormous load is taken from Asia on the search part of the site.

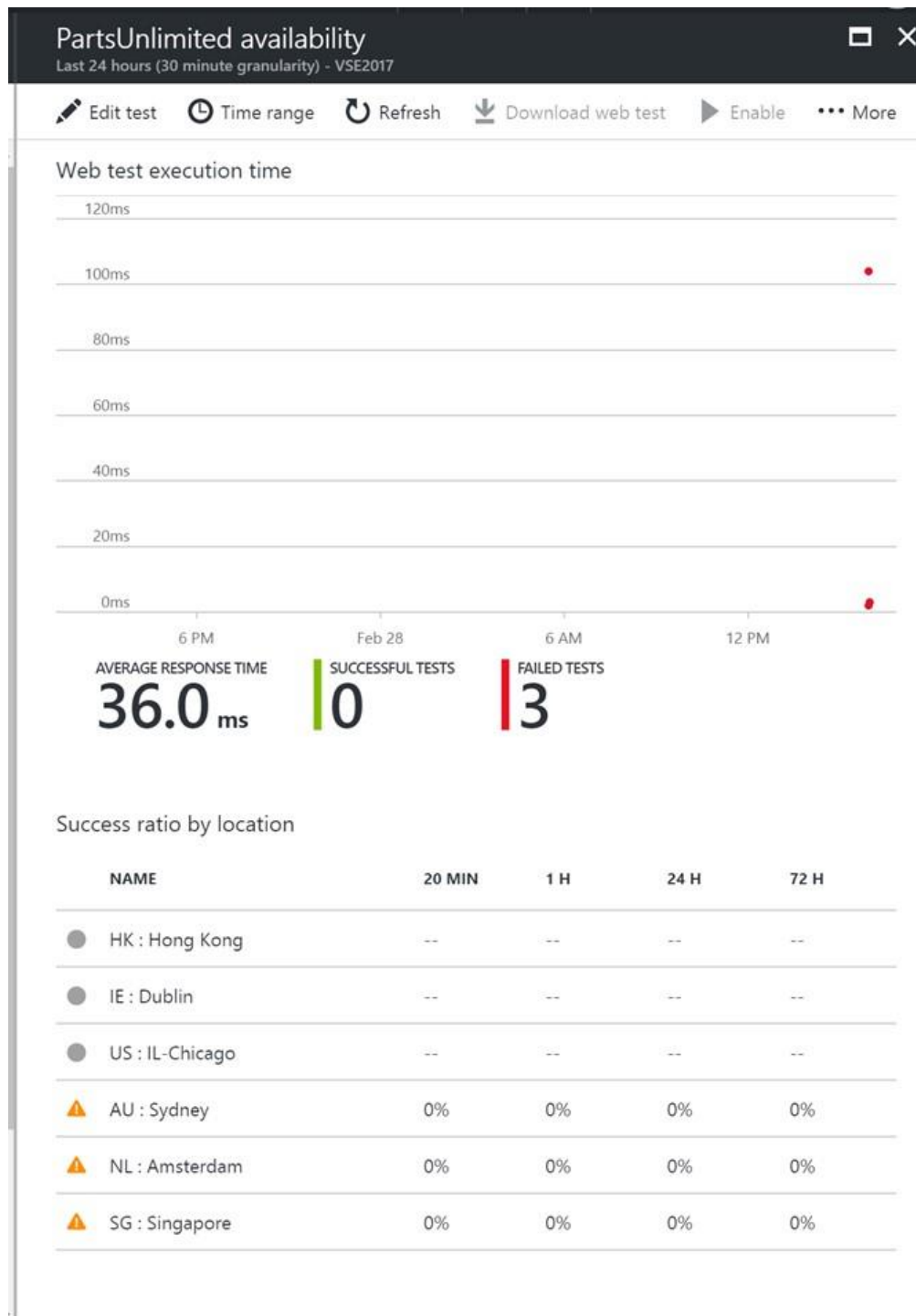
Validate Europe is still available when the Asia load is generated.

Lab 5: Set Availability Ping Tests for PartsUnlimited Website

1. Continue from creation of Application Insights resources and Azure WebApp resource in previous labs.
2. Configure availability tests and monitor availability from different regions.



- Open the Azure Portal and go to the Application Insights resource.
- Select Web Tests in the overview pane.
- Add WebTest, "PartsUnlimited availability".
- Select locations in Asia and Europe.
- Investigate the graph.

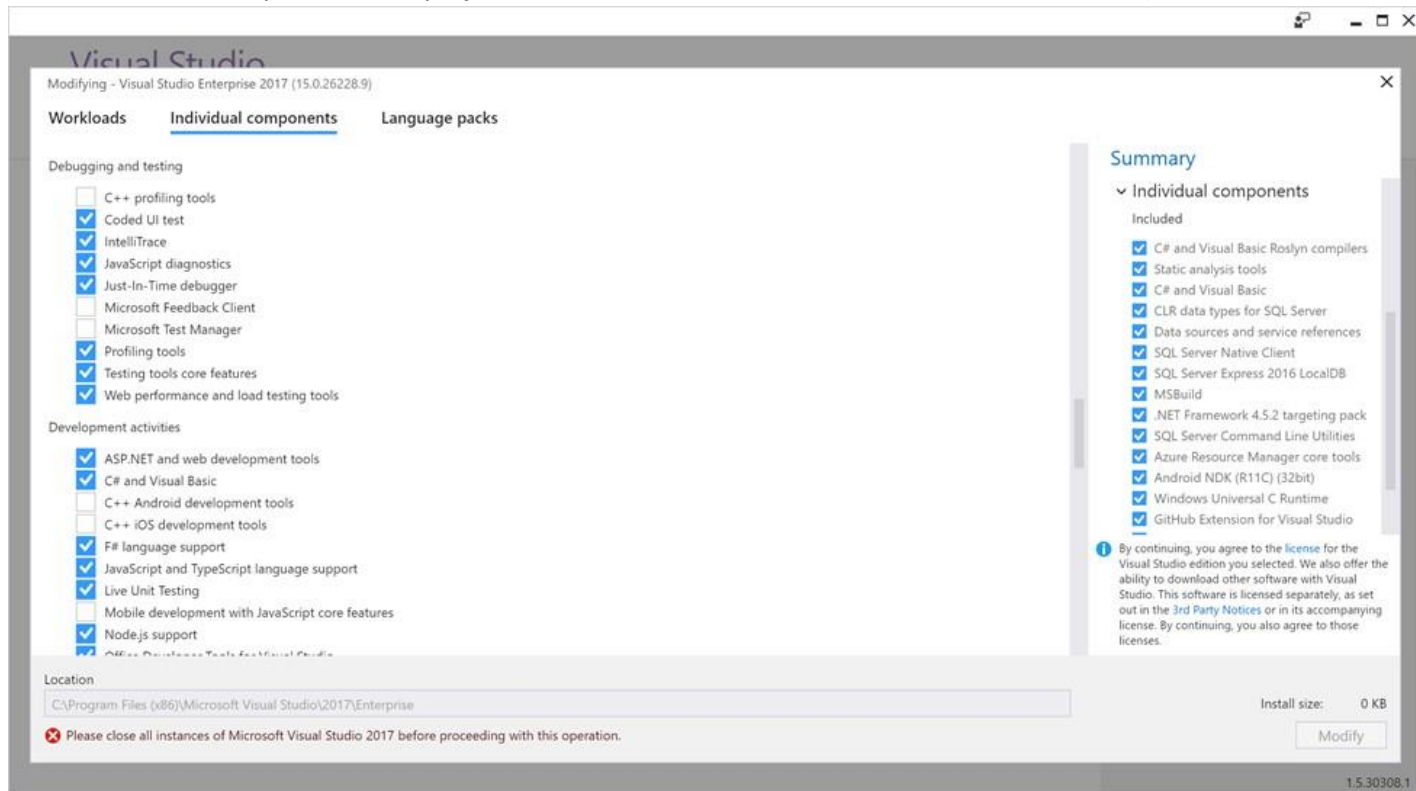


Lab 6: Create Multi-Step Web Test

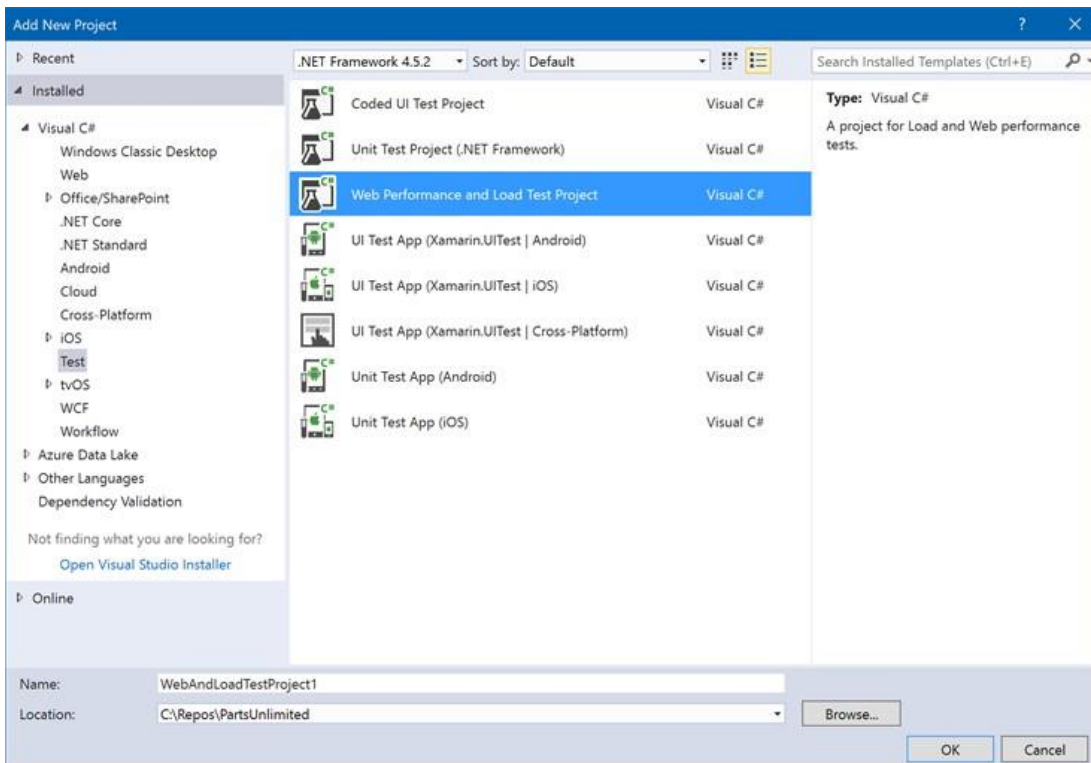
Create add to basket multi step test, with response time requirement in Visual Studio.

1. Open Visual Studio with the PartsUnlimited solution. Add a WebAndLoadTests project to the solution.

(Note: if the load and performance project isn't available, install it via the Visual Studio Installer.)

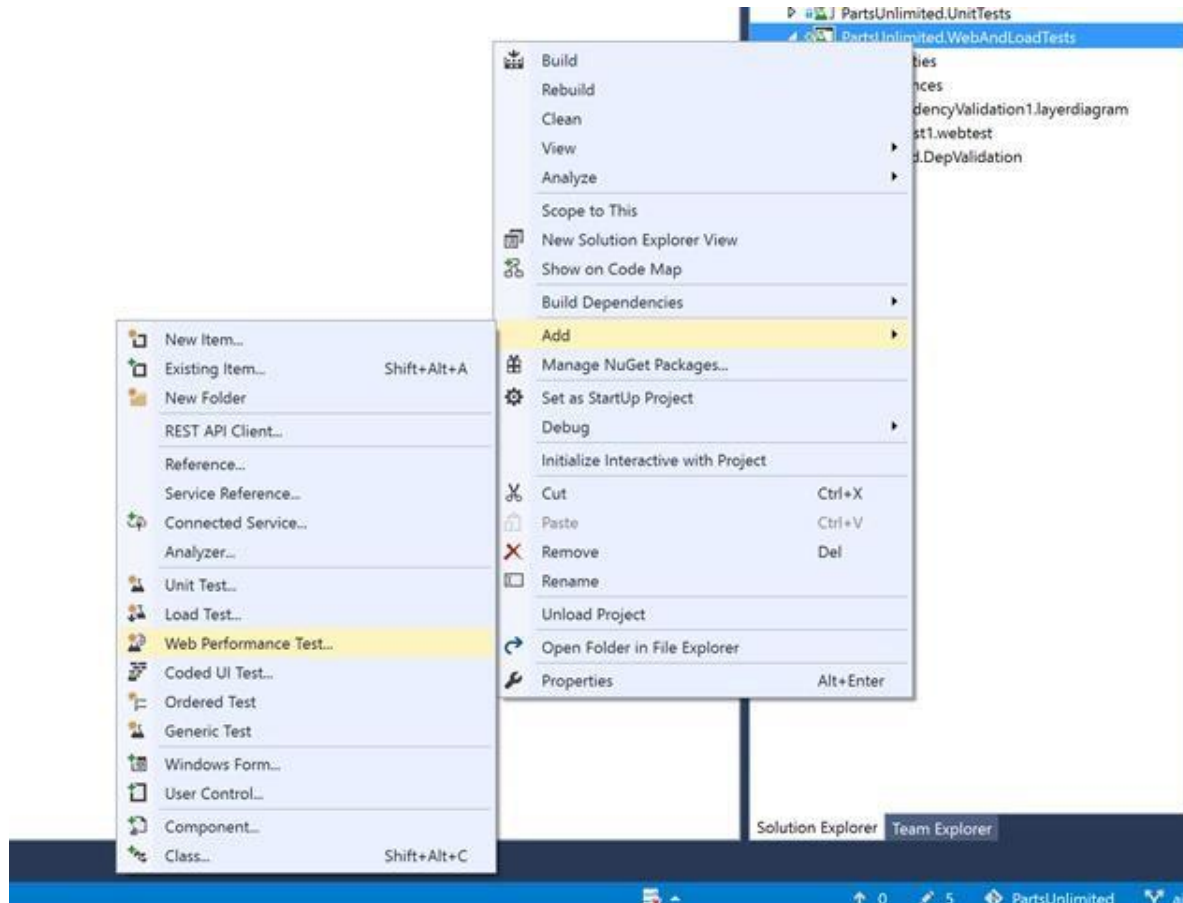


- a) Select the solution folder Tests and add a WebAndLoadTests project. Name it: PartsUnlimited.WebAndLoadTests.



2. Add a Web Performance Test to the WebAndLoadTests project.

a) Right-click on the WebAndLoadTest project and add a Web Performance Test.



b) Record the checkout scenario with the web test recorder.

- a) Open the PartsUnlimited website.
- b) Select 'jumper lead' product.
- c) Select the product.
- d) Check out.
- e) Stop the recording.
- f) Delete any requests made to any host different than *.azurewebsites.net (e.g., calls to app Insights, like dc.services.visualstudio.com) to make sure the test isn't testing any external service like Application Insights.

PartsUnlimited - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST LOAD TEST ANALYZE WINDOW HELP

WebTest3 [09:00] WebTest3.webtest WebTest1.webtest

Passed Click here to run again Internet Explorer 9.0 LAN Edit run settings

Request	Status	Total Time	Request Time	Request Bytes	Response Bytes
http://partsunlimitedvse2017demo.azurewebsites.net/	200 OK	1.429 sec	1.429 sec	0	6,390
http://partsunlimitedvse2017demo.azurewebsites.net/fonts/glyphicons-halflings-reg.	200 OK	0.339 sec	0.339 sec	0	20,127
http://partsunlimitedvse2017demo.azurewebsites.net/Search	200 OK	0.212 sec	0.212 sec	0	3,410
http://partsunlimitedvse2017demo.azurewebsites.net/fonts/glyphicons-halflings-reg.	200 OK	0.161 sec	0.161 sec	0	20,127
http://partsunlimitedvse2017demo.azurewebsites.net/Store/Details/15	200 OK	0.196 sec	0.196 sec	0	3,764
http://partsunlimitedvse2017demo.azurewebsites.net/fonts/glyphicons-halflings-reg.	200 OK	0.160 sec	0.160 sec	0	20,127
http://partsunlimitedvse2017demo.azurewebsites.net/Recommendations/GetRecommen	200 OK	0.298 sec	0.298 sec	0	0
http://partsunlimitedvse2017demo.azurewebsites.net/ShoppingCart/AddToCart/15	302 Found	0.447 sec	0.204 sec	0	130
http://partsunlimitedvse2017demo.azurewebsites.net/ShoppingCart	200 OK	-	0.243 sec	0	4,825
http://partsunlimitedvse2017demo.azurewebsites.net/fonts/glyphicons-halflings-reg.	200 OK	0.158 sec	0.158 sec	0	20,127
http://partsunlimitedvse2017demo.azurewebsites.net/Checkout/AddressAndPayment	302 Found	0.545 sec	0.172 sec	0	0
http://partsunlimitedvse2017demo.azurewebsites.net/Account/Login	200 OK	-	0.373 sec	0	4,034
http://partsunlimitedvse2017demo.azurewebsites.net/fonts/glyphicons-halflings-reg.	200 OK	0.163 sec	0.163 sec	0	20,127

Web Browser Request Response Context Details

PartsUnlimited

Parts Unlimited

a Fabrikam subsidiary

Toggle navigation

- Home
- Cart
- Log in
- Categories
 - Brakes
 - Lighting
 - Wheels & Tires

Search

Solution Explorer

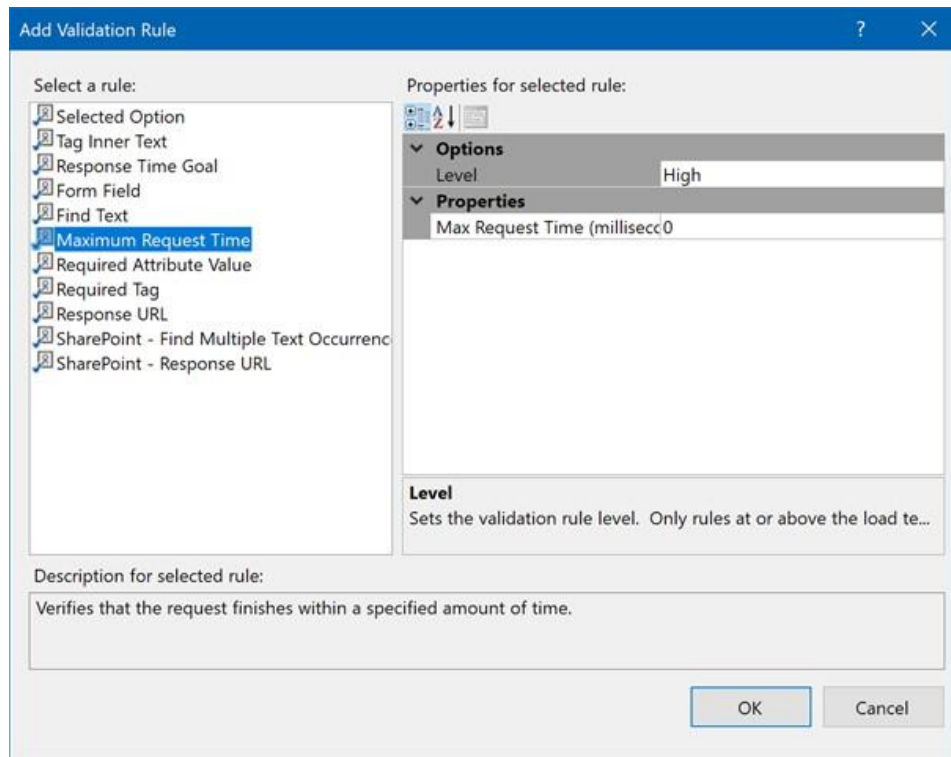
Search Solution Explorer (Ctrl+J)

Solution 'PartsUnlimited' (5 projects)

- src
 - PartsUnlimitedWebsite
 - Solution Items
 - LocalTestSettings
 - FabrikamFiber.SeleniumTests
 - PartsUnlimited.UnitTests
 - PartsUnlimited.WebAndLoadTests
 - Properties
 - References
 - packages.config
 - WebTest1.webtest
 - WebTest3.webtest
- PartsUnlimited.DepValidation
 - References
 - DependencyValidation1.LayerDiagram

Ready

- Rename the WebTest to 'CheckoutProductScenario.webtest'.
- Add a Maximum Request Time Validation to validate requests are processed within 1 second.



- e) Run the web test scenario and validate all requests are responding within 1 second. *Note: when a test step returns a 404 on a jquery request, set the property 'parse dependent requests' to false.*
3. Set the multi-step test as an availability test for Europe regions.
- Open the Azure portal and go to the Application Insights resource.
 - Select availability tests.
 - Add the CheckoutProductScenario.webtest multi-step tests.

Create test

* Test name

CheckoutScenario

✓


Test type

Multi-step test

▼

Upload a multi-step test

CheckoutProductScene



Enable retries for web test failures. ⓘ

☒

Test frequency ⓘ

5 minutes

▼


Test locations ⓘ

5 location(s) configured

>

Success criteria ⓘ

Criteria specified in test file



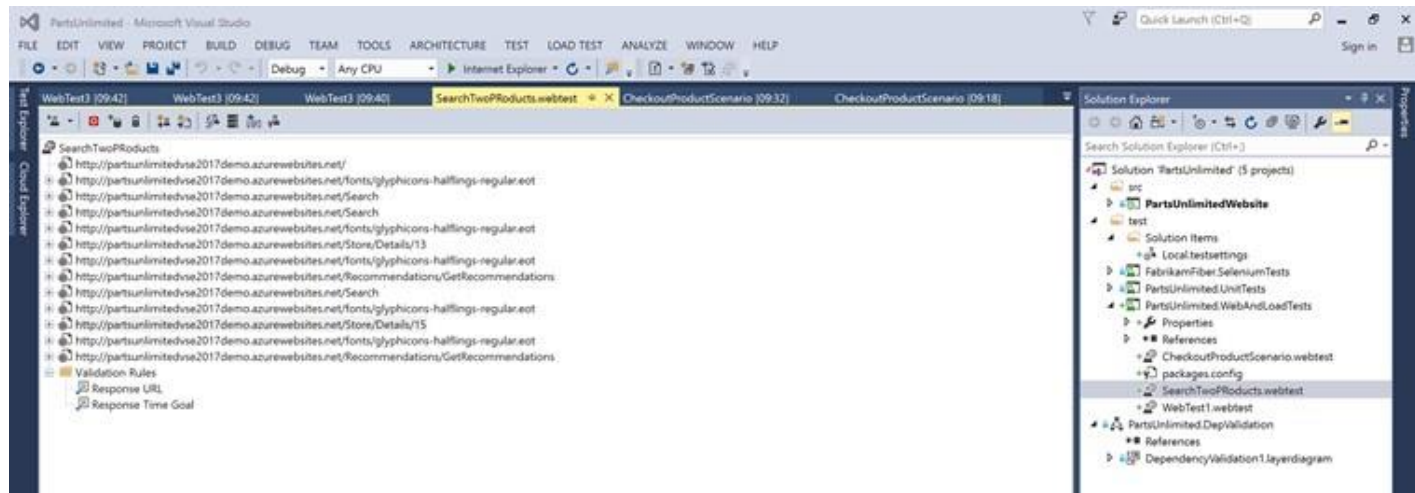
Alerts ⓘ

Alert if 3/5 locations fails in 5 min...

>

- d) Set the locations to run the test from to Europe and US locations.
- e) Configure the WebTest to send alerts to your email address.

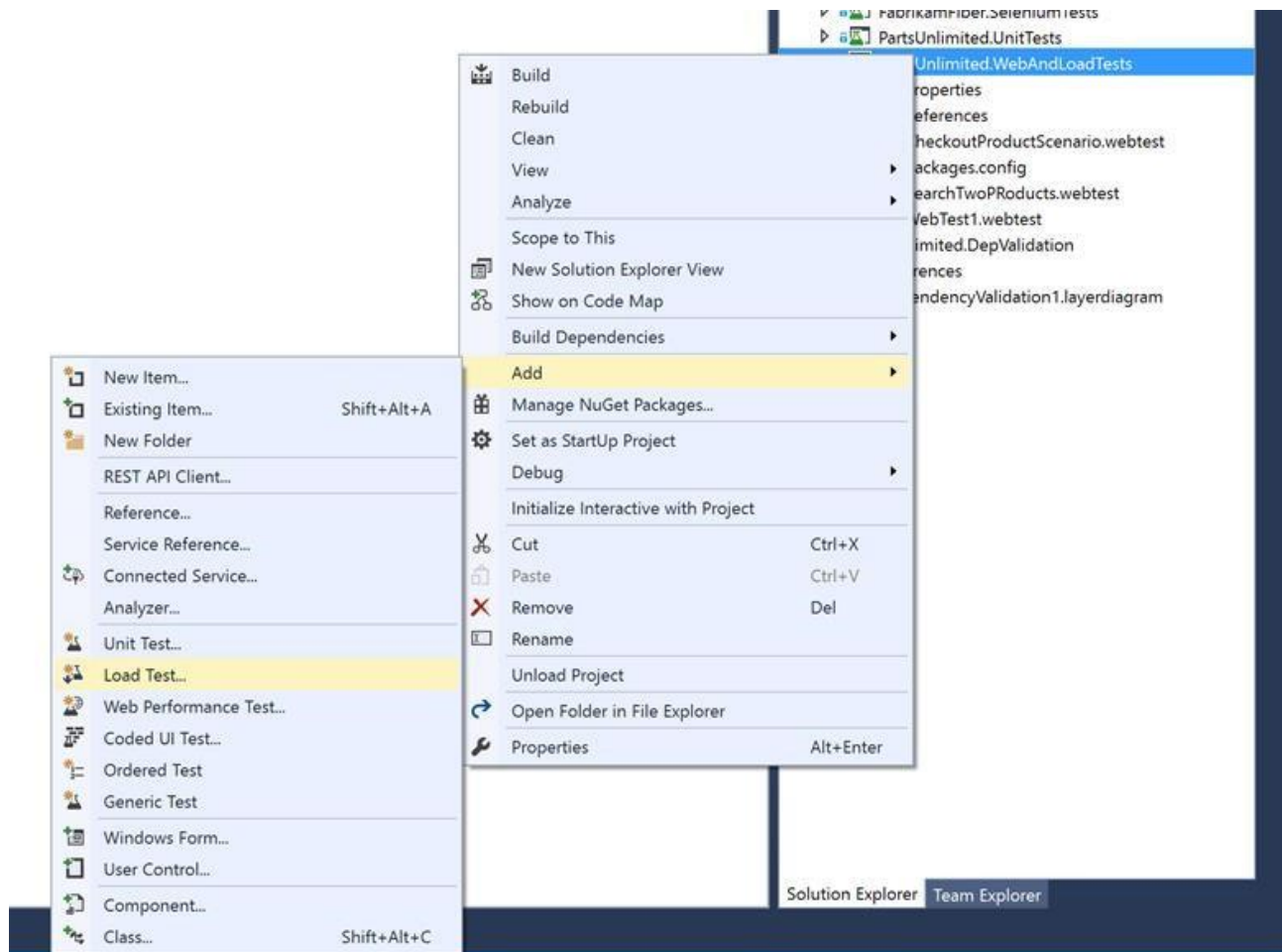
- iii. Select Battery product.
- iv. Search for Jumper Lead.
- v. Select Jumper Lead product.



- c) Delete any requests made to any host different than *.azurewebsites.net (e.g., calls to App Insights like dc.services.visualstudio.com) to make sure the test isn't testing any external service like Application Insights.
- d) Run the performance test to validate the run successes.

Note: when a test step returns a 404 on a jquery request, set the property 'parse dependent requests' to false.

- e) Save the WebPerformance test as SearchProducts.webtest.
- f) Add a Load Test to the WebAndLoadTests project.



g) Select Cloud Load Test.

New Load Test Wizard

Welcome to the Create New Load Test Wizard

This wizard helps you create a load test. Select the type of load test:

☒ **Cloud-based Load Test with Visual Studio Team Services**

- Create a performance lab in the cloud in minutes
- Generate high user load from any Azure datacenter
- Get free user minutes every month. [Learn more](#)

Accou...

☐ **On-premise Load Test**

- Use your existing on-premises performance lab
- Test with minimal latency

Your selection will be applied to the file: local.testsettings

< Previous Next > Finish Cancel

- h) Create load test for 10.000 users for 10 minutes. Keep all settings in the New Load Test Wizard to default except:
- Set load test run settings to 10 minutes.

New Load Test Wizard

Review and edit run settings for a load test

Specify the length of the load test by:

☒ **Load test duration**

Warm-up duration (hh mm ss):

Run duration (hh mm ss):

☐ **Test iterations**

Test iterations:

Details

Sampling rate: seconds

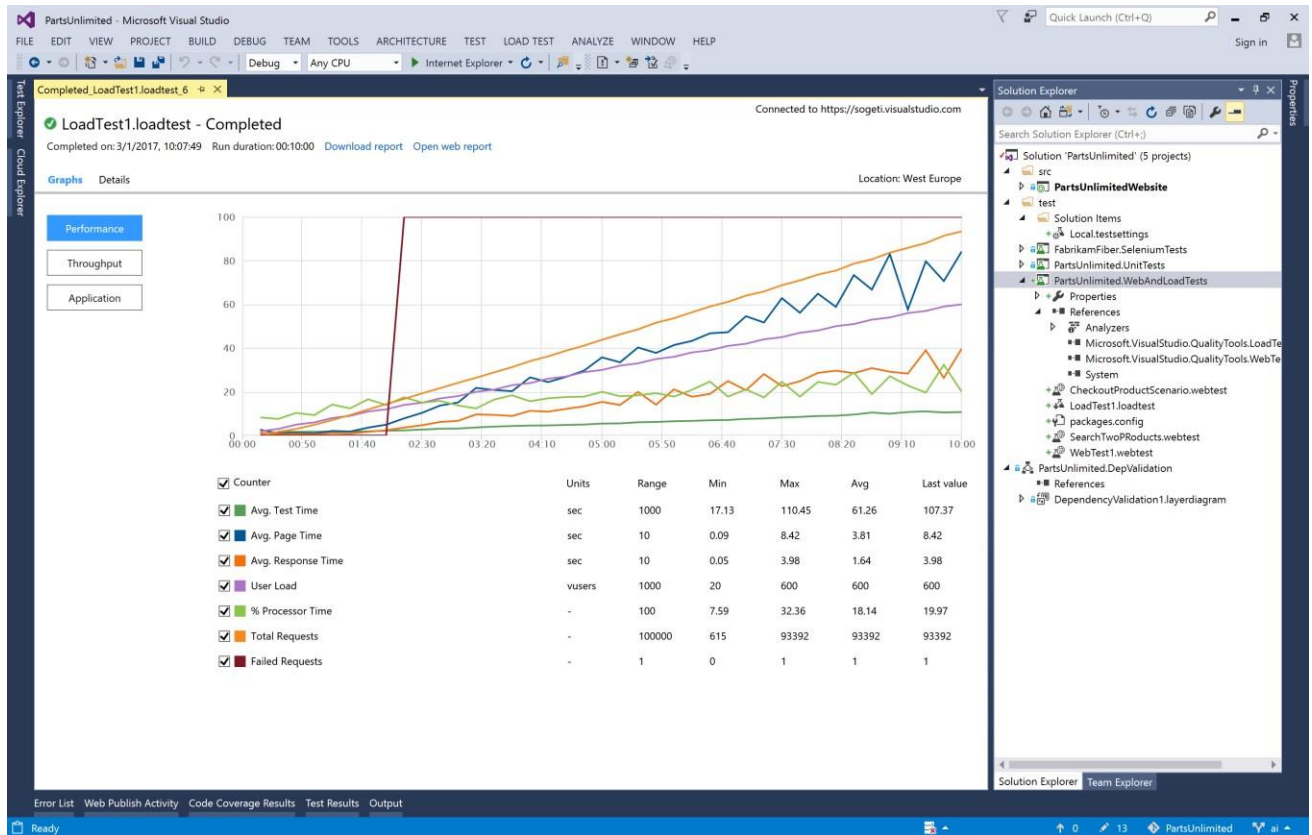
Description:

Save Log on Test Failure:

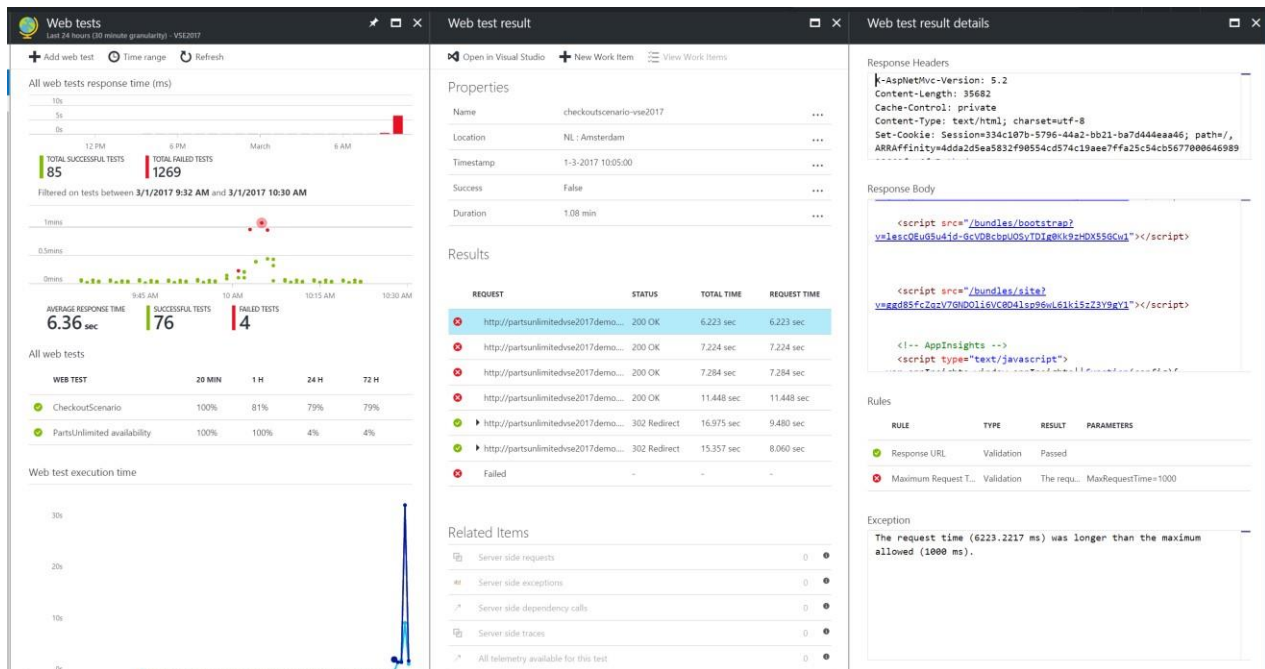
Validation level:

< Previous Next > Finish Cancel

- Set Load Pattern to 'Step Load' maximum user count 10.000



- k) Investigate the Availability tests. Focus on the failing test during the load test. This indicates the checkout process in the US and EU are influenced by the search activity load.



- l) Investigate mailbox on send alerts from the WebTest.

Reply Reply All Forward



App Insights Alerts <ai-noreply@microsoft.com>

Clemens Reijnen

01

Alert: "Average Server exceptions" for app "VSE2017"

1 If there are problems with how this message is displayed, click here to view it in a web browser.

Bing Maps

+ Get more add-ins



Azure Application Insights



WARNING

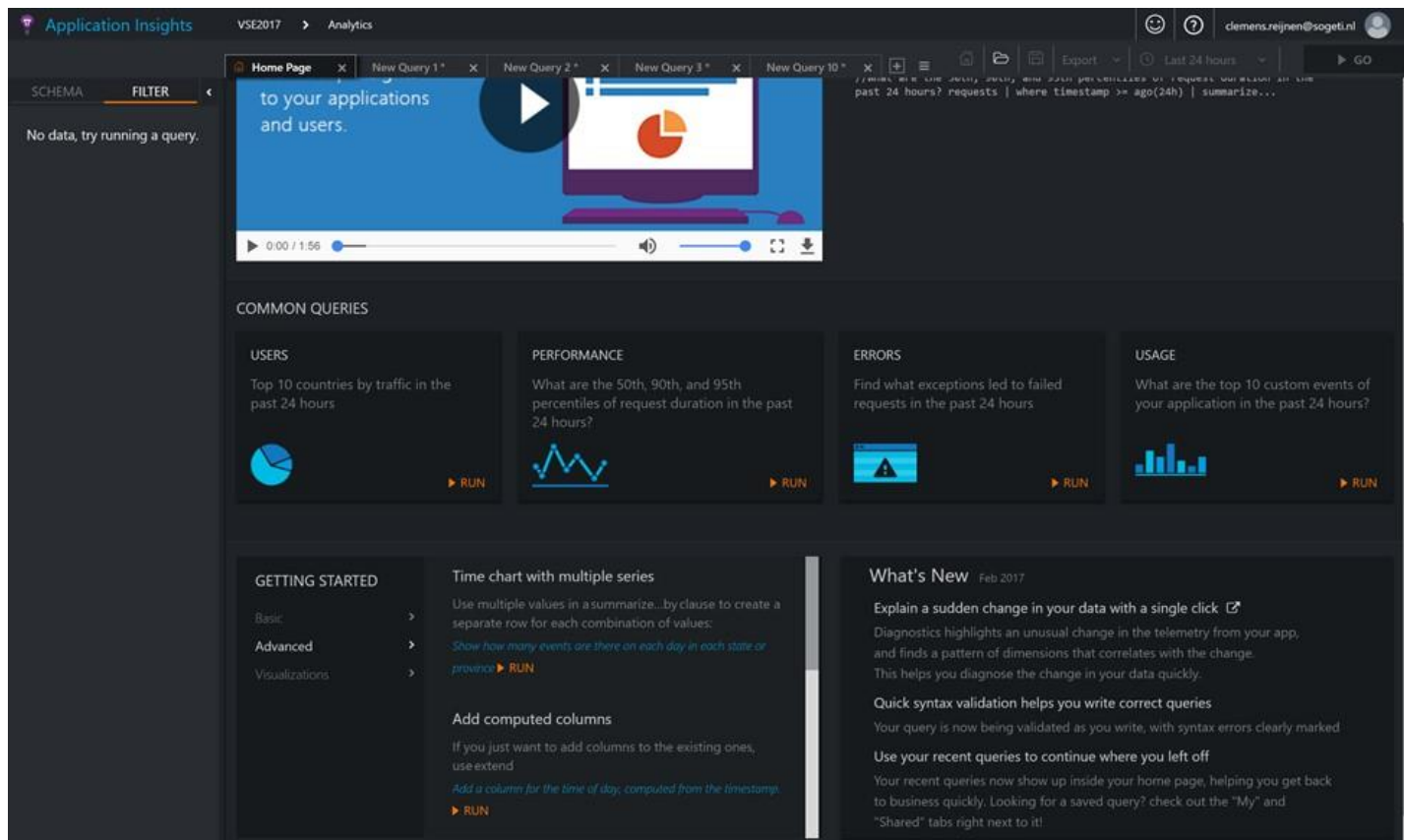
Average Server exceptions is 2 : crossed threshold for 5 minutes

[View application in Azure Portal](#)

Time	03/07/2017 23:44:31 (UTC)
Application	VSE2017
Subscription	MSFT Product Development
Performance metric	Average Server exceptions
Configured time period	5 minutes
Configured condition	>
Configured threshold	1

Lab 8: Log Analytics with Application Insights

1. Open Azure portal.
2. Go to the created Application Insights resource.
3. Select 'Analytics' in the overview top-menu bar.
4. Investigate the query language capabilities.
5. Select 'Getting Started' → 'Advanced' → 'Time chart with multiple series'.



6. Open a new query screen.

7. Add the following query to get the top 10 common page flows for my users:
pageViews

```
| WHERE TIMESTAMP >= AGO(1D)
| ORDER BY TIMESTAMP DESC
| SUMMARIZE PAGESVISITED=MAKELIST(NAME) BY SESSION_ID
| SUMMARIZE COUNT() BY TOSTRING(PAGESVISITED)
| TOP 20 BY COUNT_DESC
```

Application Insights VSE2017 > Analytics

Home Page x New Query 1* x

Filter by Name or Type...

COLLAPSE ALL | EXPAND ALL

APPLICATION INSIGHTS

- traces
- customEvents
- pageViews
- requests
- dependencies
- exceptions
- availabilityResults
- customMetrics
- performanceCounters
- browserTimings

OTHER DATA SOURCES

Apply Analytics to your own data. [Contact us](#)

```
//What are the top 10 common page flows for my users
pageViews
| where timestamp >= ago(1d)
| order by timestamp desc
| summarize pagesVisited=makeList(name) by session_Id
| summarize count() by toString(pagesVisited)
| top 20 by count_desc
```

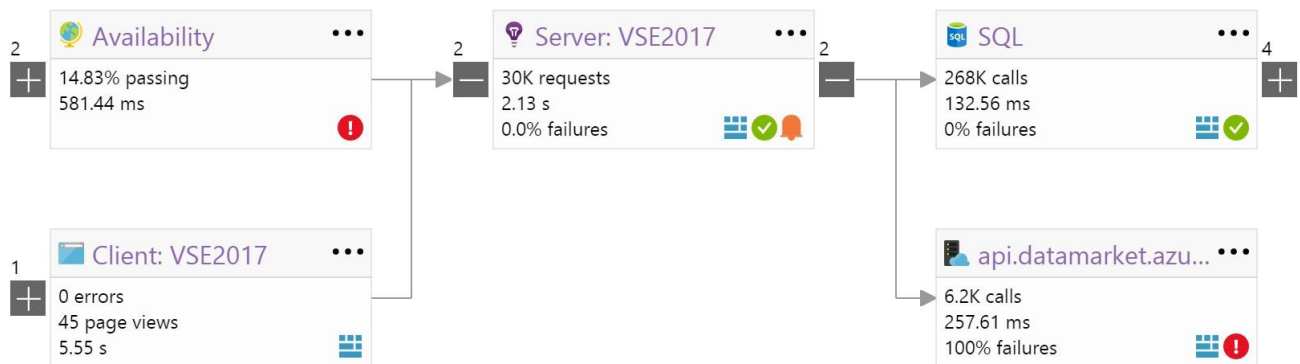
Completed 00:00:01.070 5 records loaded

Drag a column header and drop it here to group by that column

pagesVisited	count_
["Home Page - Parts Unlimited"]	2
["- Parts Unlimited", "- Parts Unlimited", "Home Page ..."]	1
["Shopping Cart - Parts Unlimited", "Product - Jumper..."]	1
["Home Page - Parts Unlimited", "Home Page - Parts Un..."]	1
["Store - Parts Unlimited", "Home Page - Parts Unlimi..."]	1

Lab 9: Application Map

1. Go to the Azure Application Insights Overview Blade, click on the Application Map button to open the Application Map. Investigate the application Map. Dependencies are automatic capture, client/server side components, and SQL server components, along with status and basic performance metrics of each component.
2. Click on any red/orange warning symbols to open actual failure details to can click through and investigate.





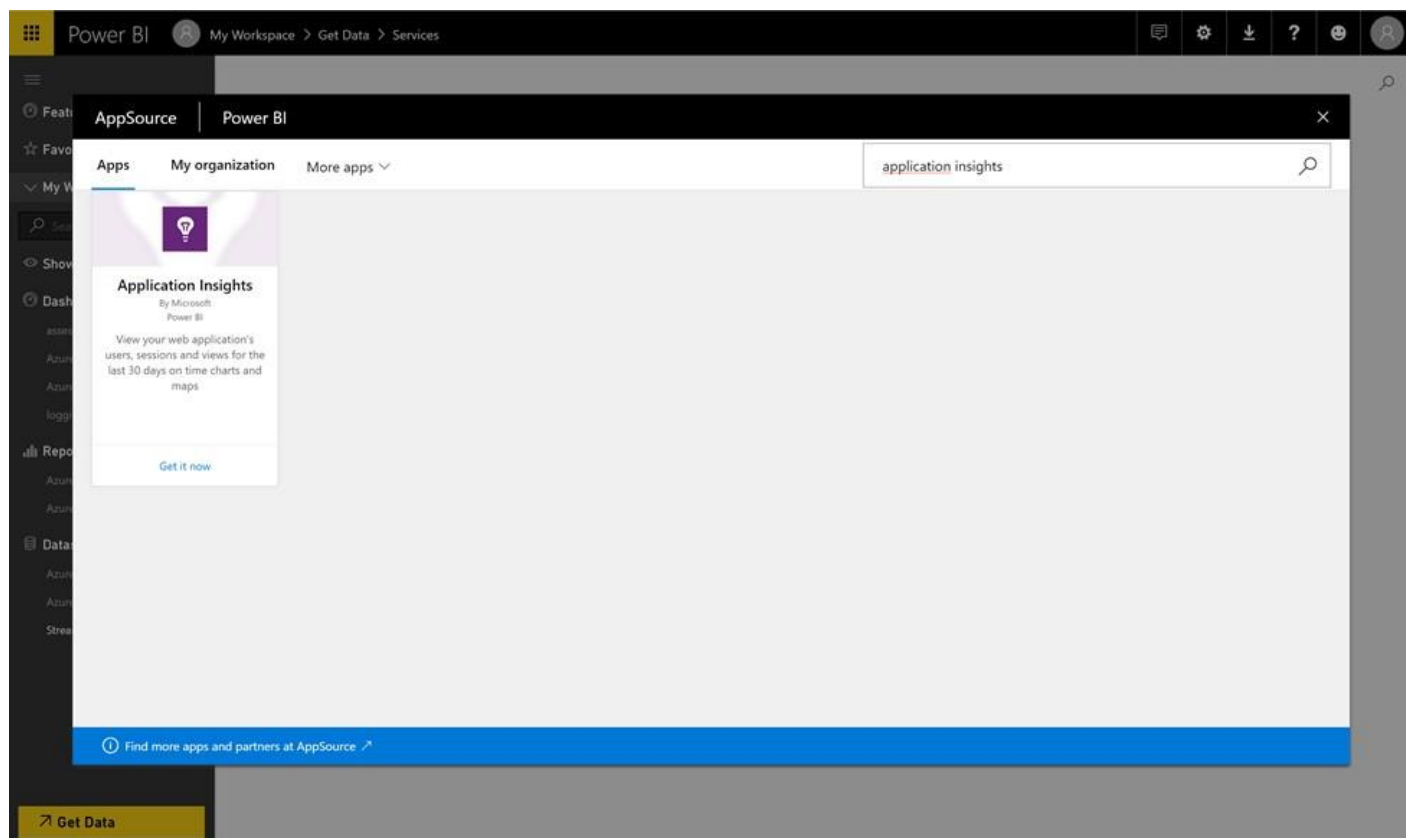
Scenario 4: Customize Reports and Add Custom Telemetry Data

The product owner and sales colleagues who are responsible for the PartsUnlimited website and sales want to access to the usages telemetry data so they can investigate the pages visited and make queries and reports themselves.

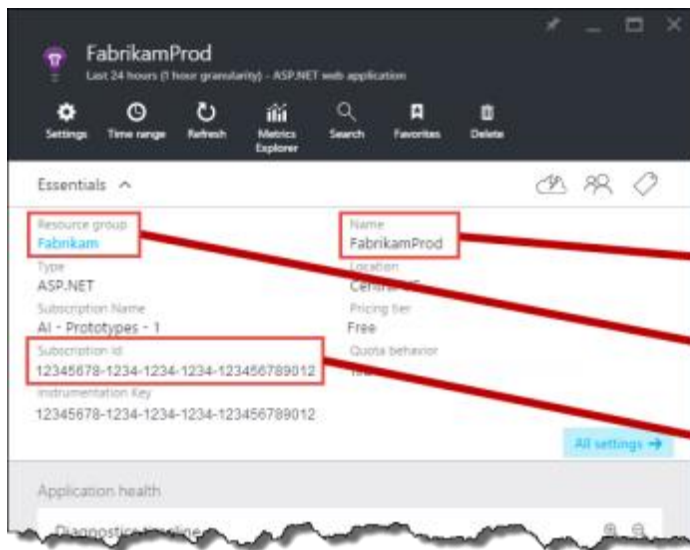
As a PartsUnlimited team member, enable Application Insights usages data to be available in PowerBI.

Lab 10: Telemetry Data in Power BI

1. Sign up for PowerBI or sign in to an existing PowerBI instance. Follow the steps on: <https://powerbi.microsoft.com/en-us/documentation/powerbi-service-self-service-signupfor-power-bi/> 2. Add PowerBI.



3. Authenticate PowerBI to access Application Insights.



Connect to Application Insights

To start using your Application Insights data in Power BI, follow the prompts below.
Need help connecting? [Learn more](#)

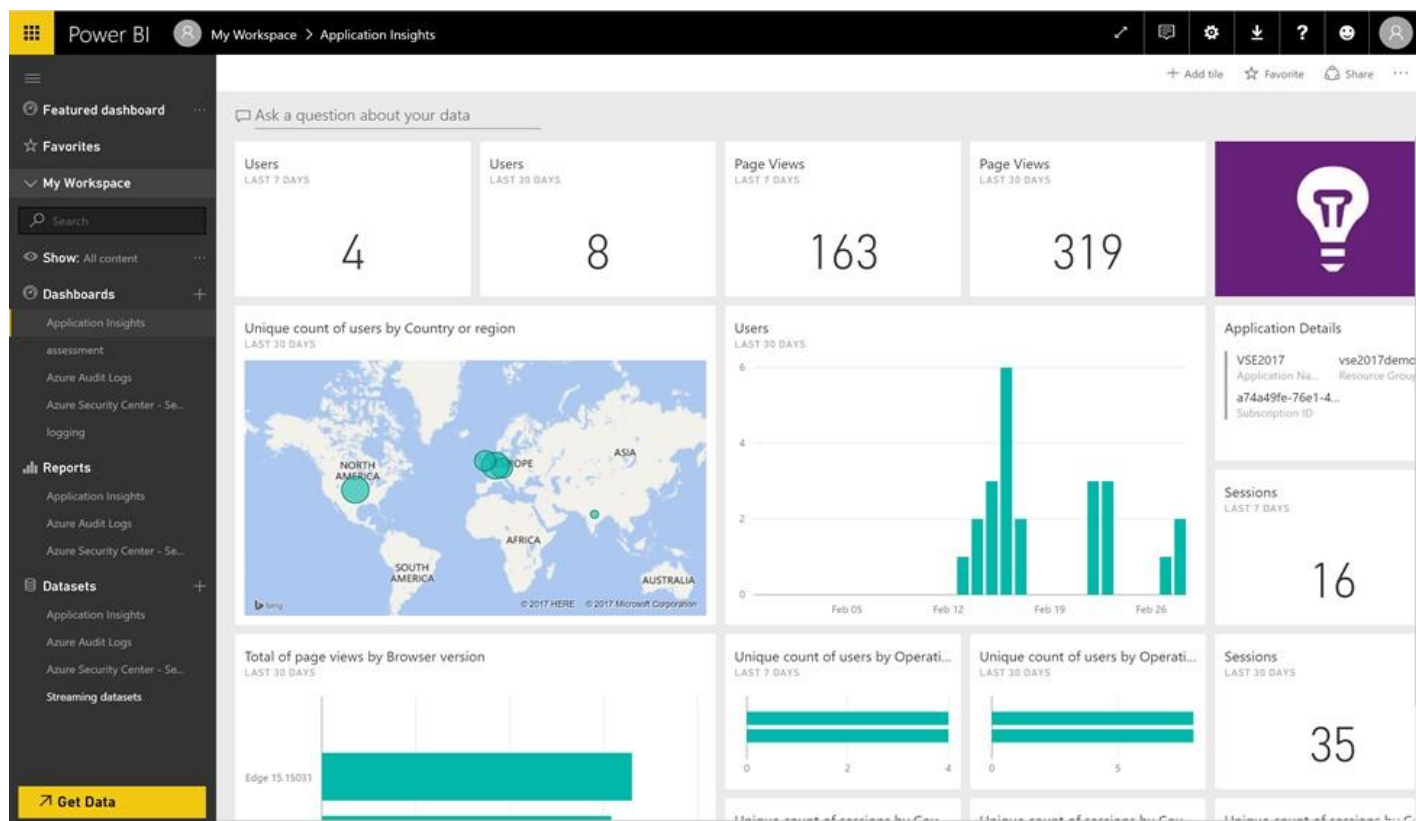
Application Insights Resource Name
Copy from the Essentials dropdown on the the resources blade in the Azure Portal
Example: MyApp

Resource Group
Resource group containing the Application Insights resource
Example: MyResourceGroup

Azure Subscription ID
ID of the subscription containing Application Insights resource
Example: 00000000-0000-0000-0000-000000000000

[Next](#) [Cancel](#)

4. Investigate the created dashboard.





Scenario 5: OMS

Now that the sales and product owners of the PartsUnlimited website have their own view and query capabilities in their own tool, the DevOps team also wants to have their own dashboard with insights data, which is customizable for their own needs and within their own tool.

As a DevOps team member, configure Operations Management Suite to visualize Application Insights data from the PartsUnlimited website.

Lab 11: Setup and OMS with Application Insights

1. Setup and OMS Workspace.
 - a) Navigate to <http://microsoft.com/oms> and click the Try for free button. Sign in with your Microsoft account such as Outlook.com, or with an Organizational account provided by your company or educational institution to use with Office 365 or other Microsoft services.
 - b) Provide a unique Workspace Name. A workspace is a logical container where your management data is stored. It provides you a way to partition data between different teams in your organization, as the data is exclusive to its workspace. Specify an email address and the region where you would like to have your data reside.
 - c) Next, you can create a new Azure subscription or link to an existing Azure subscription.
2. Add the Application Insights Collector to the OMS workspace.

The screenshot displays the Microsoft Operations Management Suite (OMS) interface. On the left, the 'Solutions Gallery' shows the 'Application Insights Connector (Preview)' as 'Available' with an 'Add' button. The description states: 'The Application Insights Solution provides a way to connect multiple applications to your Log Analytics workspace. With your infrastructure data (performance counters, configuration changes, updates, etc) and application data in one place, you can better investigate problems across your stack. With this solution you will be able to: • Connect your existing applications from Application Insights to Log Analytics • Explore Requests, Exceptions, Availability, Page Views, and Custom Events telemetry types in Log Analytics Search and solution views'.

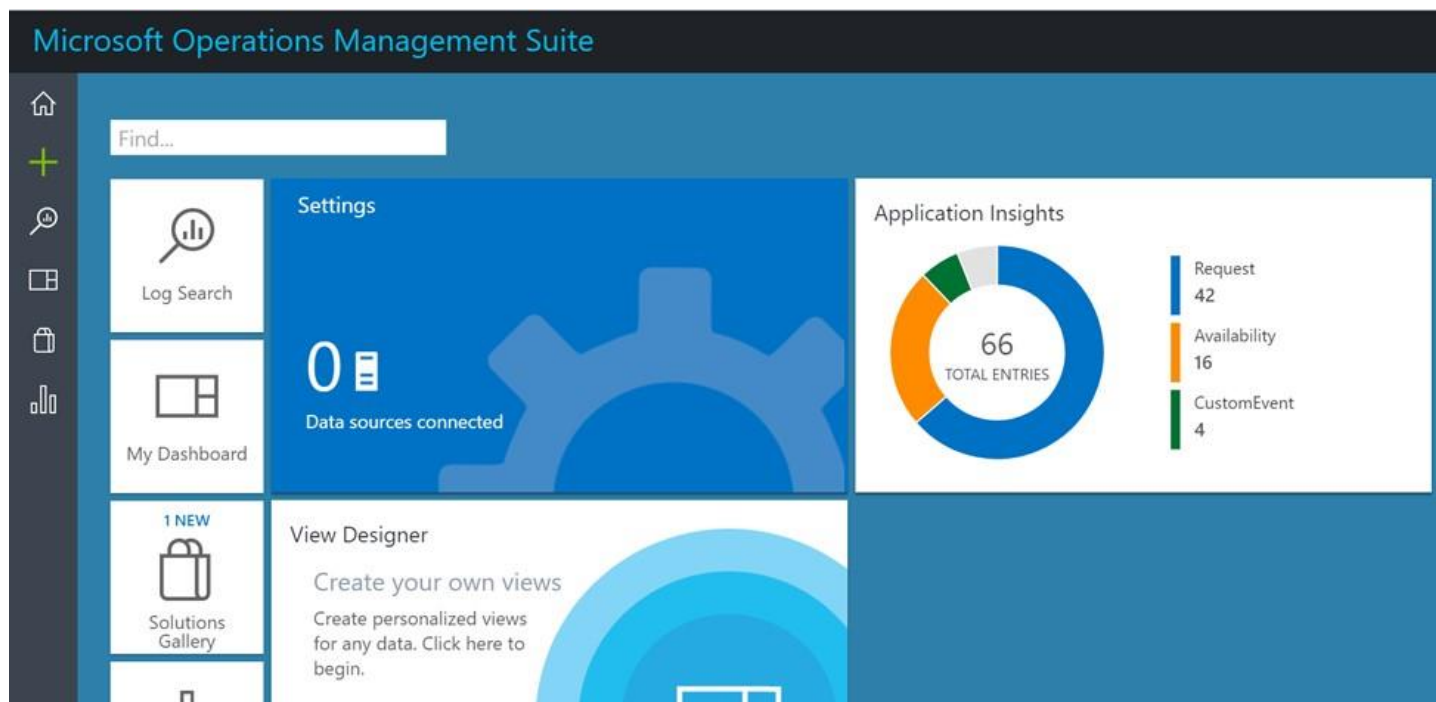
The main dashboard, titled 'Microsoft Operations Management Suite Overview > Application Insights', provides a comprehensive view of application health. It includes sections for 'APPLICATIONS' (showing 5 applications), 'DATA VOLUME' (9 flows sending data), 'AVAILABILITY' (75.5K requests, 49 failed), and 'SERVER REQUESTS' (Requests per hour graph). Below these are detailed tables for applications, computers, and failed requests.

APPLICATIONS	ISSUE	COMPUTER	COUNT
chrisgopal-prd	728.0K	Microsoft Azure cloudapp...	501.0K
Microsoft Dynamics 365	814.0K	Microsoft Dynamics 365...	101.0K
Microsoft Dynamics 365	271.0K	Microsoft Dynamics 365...	20.0K
Microsoft Dynamics 365	103.0K	Microsoft Dynamics 365...	10.0K
Microsoft Dynamics 365	30.0K	Microsoft Dynamics 365...	10.0K

APPLICATIONS	ISSUE
Microsoft Dynamics 365	10.0K
Microsoft Dynamics 365	10.0K
Microsoft Dynamics 365	10.0K
Microsoft Dynamics 365	10.0K
Microsoft Dynamics 365	10.0K

APPLICATIONS	ISSUE
Microsoft Dynamics 365	10.0K
Microsoft Dynamics 365	10.0K
Microsoft Dynamics 365	10.0K
Microsoft Dynamics 365	10.0K
Microsoft Dynamics 365	10.0K

3. Configure the Application Insights Connector by linking the PartsUnlimitedWebsite Azure Resource.



4. Click on the Application Insights widget to see the charts and metrics right within OMS.

Lab 12: Add SQL Data to the OMS Workspace

1. Open the OMS Workspace created in Lab 11.
2. Add the Azure SQL Analytics solution to the workspace.

Azure SQL Analytics (Preview)
Owned

[View](#) [Remove](#)

Description

Operations Management Suite (OMS) is a management offering which provides monitoring for Azure Resources through the Log Analytics service. Log Analytics enables users to collect, correlate and visualize structured and unstructured data. Through the out of the box solutions available in OMS Log Analytics, users can easily monitor and receive notifications on the health of their Azure Resources such as SQL Azure.

Microsoft Azure SQL Database, also known as Azure SQL, is a scalable relational database service that provides familiar SQL-Server-like capabilities to applications running in Azure cloud. OMS Log Analytics collects and visualizes the important SQL Azure performance metrics and enables users to easily create custom monitoring rules in addition to those provided with the solution. OMS Log Analytics enables you to monitor across multiple Azure subscriptions, resources and elastic pools and more importantly lets you identify issues at each layer of your application stack. This preview solution supports up to 150,000 Azure SQL Databases and 5,000 SQL Elastic Pools.

If you use this solution, you will need to enable Azure SQL Metric logging by following the instruction found here: <https://aka.ms/sqlazureonboarding>

Azure SQL DATABASE ANALYTICS

Top N Databases by DTU Utilization > 90%

SQL DATABASE NAME	AVERAGE % DTU
CONTOSOESTDB	83.6
BACONANDEGGS	6.9
BEERANDPRETZELS	6.8
CHEETOS	6.8
CHORIZO	6.7
SATYAVELAZURESQLDB	6.7
SATYAVELSQLAZDB	6.5
BACONANDBEER	5.8
ANGELTESTDB	0
CONTACTMANAGERANGE...	0

See all...

Top N Databases by CPU Utilization > 90%

SQL DATABASE NAME	AVERAGE % CPU
CONTOSOESTDB	64.2
BACONANDEGGS	7.2
BEERANDPRETZELS	7.1
CHEETOS	7
SATYAVELAZURESQLDB	7
CHORIZO	7
SATYAVELSQLAZDB	6.8
BACONANDBEER	6.1
ANGELTESTDB	0
CONTACTMANAGERANGE...	0

See all...

Top N Databases by St...

SQL DATABASE NAME
ANGELTESTDB
BACONANDBEER
BACONANDEGGS
BEERANDPRETZELS
CHEETOS
CHORIZO
CONTOSOESTDB
REDTSHIRT
SATYAVELAZURESQLDB
SATYAVELSQLAZDB

See all...

3. Configure OMS to monitor the PartsUnlimited SQL Server (See also steps in this [blog](#)).
 - a) Install the latest set of AzureRM modules on your workstation: PS C:\> install-module -

Name AzureRM -Force

- b) Open PowerShell in Administrator mode: Run as Administrator.
- c) Save the Enable-AzureRMDiagnostics.ps1 script file locally, run the following command, and provide a path to store the script:
PS C:\> save-script -Name Enable-AzureRMDiagnostics -Path
"C:\users\\desktop\temp"
- d) Go to the folder where you saved the script, and execute Enable-AzureRMDiagnostics.ps1. PS C:\users\\Desktop\temp> .\Enable-AzureRMDiagnostics.ps1

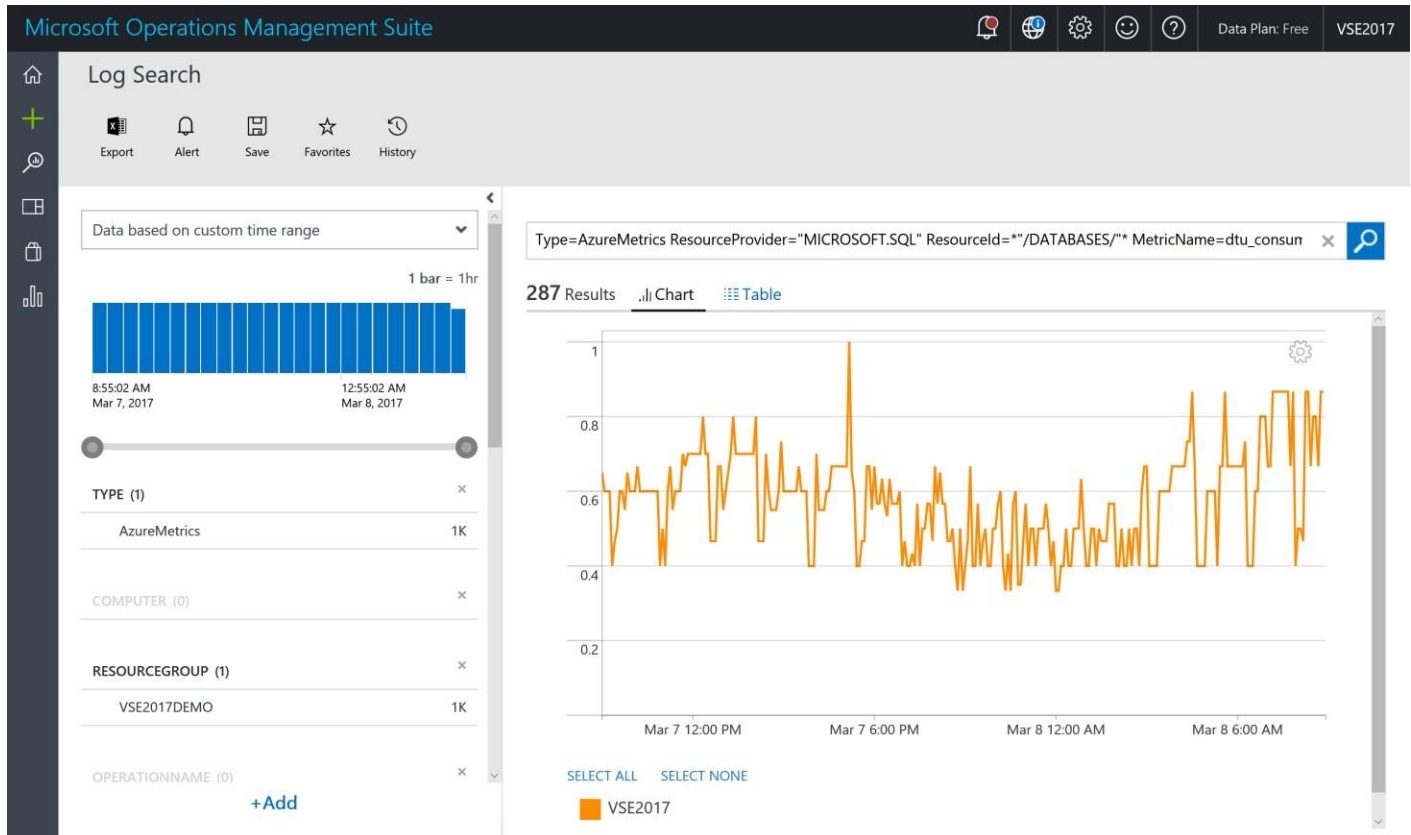
Note: Narrow down the monitoring to PartsUnlimited with the resource group switch in the command -ResourceGroup "PartsUnlimited-RG"

4. Configure alerting on high DTU consumption.

- a) Select, in the list of popular SQL queries, "Alert on High DTU Consumption for SQL Databases"

LIST OF POPULAR AZURE SQL SEARCH QUERIES	
All Azure SQL Data	Type=AzureMetrics ResourceProvider="MICROSOFT.SQL"
List of all Azure SQL Database Performance Metrics	Type=AzureMetrics ResourceProvider="MICROSOFT.SQL" Resour...
List of all Azure SQL Elastic Pool Performance Metrics	Type=AzureMetrics ResourceProvider="MICROSOFT.SQL" Resour...
SQL Database DTU Consumption over time	Type=AzureMetrics ResourceProvider="MICROSOFT.SQL" Resour...
SQL Elastic Pool DTU Consumption over time	Type=AzureMetrics ResourceProvider="MICROSOFT.SQL" Resour...
Alert on High DTU Consumption for SQL Elastic Pool	Type=AzureMetrics ResourceProvider="MICROSOFT.SQL" Resour...
Alert on High Database Reads on Azure SQL Databases	Type=AzureMetrics ResourceProvider="MICROSOFT.SQL" Resour...
Alert on High CPU Percent for Azure SQL Databases	Type=AzureMetrics ResourceProvider="MICROSOFT.SQL" Resour...
Alert on High DTU Consumption for Azure SQL Databases	Type=AzureMetrics ResourceProvider="MICROSOFT.SQL" Resour...

- b) Select the Alert option to configure the alert.



c) Set the required fields.

Microsoft Operations Management Suite

Log Search Add Alert Rule

General

Alert information

Name DTU Database

Description

Severity Critical

Search query Use current search query

Type=AzureMetrics ResourceProvider="MICROSOFT.SQL" ResourceId="/" DATABASES/" MetricName=dtu_consumption_percent | measure Avg(Average) by Resource interval 5minutes

Time window

Save Cancel

Schedule

Alert frequency Check for this alert every 15 Minutes

Generate alert based on Number of results Metric measurement

Number of results Greater than 1

☐ Suppress alerts When checked, allows you to set an amount of time to wait before alerting again to reduce alert noise

Actions

Email notification Yes No

Subject Email subject

Recipients (semi-colon separated) clemens.reijnen@sogeti.nl

Webhook Yes No

Runbook Yes No

