Microsoft

AZ-203.5
Module 03: Instrument solutions to support monitoring and logging

Prashanth Kumar

Microsoft

Topics

- Configure instrumentation in an app or service
- Analyze and troubleshoot solutions by using Azure Monitor

Lesson 01: Configure instrumentation in an app or service

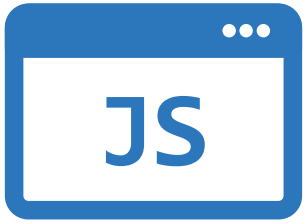
Application Insights for webpages

- Monitor webpage or applications
  - Observe usage in near real-time
  - Gather performance metrics
- Can use a script to capture front-end telemetry
  - Page load time
  - Asynchronous JavaScript and XML (AJAX) calls
  - Browser exceptions
  - AJAX failures
  - User information
  - Session counts
- Segmented breakdowns
  - By users, page, client OS, browser version, geo-location, or other dimensions

Application Insights for webpages - code

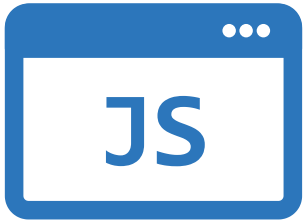
```
<script type="text/javascript">
var appInsights=window.appInsights||function(a){
function b(a){c[a]=function(){var b=arguments;c.queue.push(function(){c[a].apply(c,b)})}}var
c={config:a},d=document,e=window;setTimeout(function(){var
b=d.createElement("script");b.src=a.url||"https://az416426.vo.msecnd.net/scripts/a/ai.0.js",d.getEl
ementsByTagName("script")[0].parentNode.appendChild(b)});try{c.cookie=d.cookie}catch(a){}c.queue=[]
;for(var
f=["Event","Exception","Metric","PageView","Trace","Dependency"];f.length;)b("track"+f.pop());if(b(
"setAuthenticatedUserContext"),b("clearAuthenticatedUserContext"),b("startTrackEvent"),b("stopTrack
Event"),b("startTrackPage"),b("stopTrackPage"),b("flush"),!a.disableExceptionTracking){f="onerror",
b("_"+f);var g=e[f];e[f]=function(a,b,d,e,h){var
i=g&&g(a,b,d,e,h);return!0!==i&&c["_"+f](a,b,d,e,h),i}}return c
}({ instrumentationKey:"<your instrumentation key>" });

window.appInsights=appInsights,appInsights.queue&&0===appInsights.queue.length&&appInsights.trackPa
geView();
</script>
```

Instrumentation Key goes here



Application Insights for web pages - config

```
// Send telemetry immediately without batching.
// Remember to remove this when no longer required, as it can affect browser performance.
enableDebug: boolean,
// Don't log browser exceptions.
disableExceptionTracking: boolean,
// Don't log ajax calls.
disableAjaxTracking: boolean,
// Limit number of Ajax calls logged, to reduce traffic.
maxAjaxCallsPerView: 10, // default is 500
// Time page load up to execution of first trackPageView().
overridePageViewDuration: boolean,
// Set dynamically for an authenticated user.
accountId: string,
```

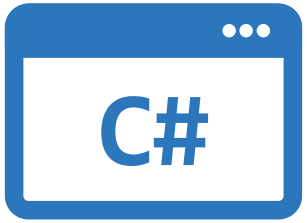


Application Insights for console applications

- Install latest **Microsoft.ApplicationInsights** NuGet package
- Install latest **Microsoft.ApplicationInsights.DependencyCollector** NuGet package
- Set the instrumentation key
  - By using the static **TelemetryConfiguration.Active.InstrumentationKey** property
  - By using the **APPINSIGHTS_INSTRUMENTATIONKEY** environment variables
  - By using the **ApplicationInsights.config** file

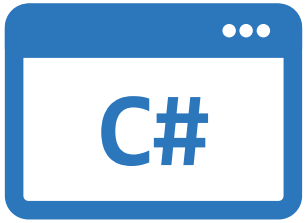
Application Insights for console applications - config

```
TelemetryConfiguration.Active.InstrumentationKey = " *your key* ";

var telemetryClient = new TelemetryClient();

telemetryClient.TrackTrace("Hello World!");
```



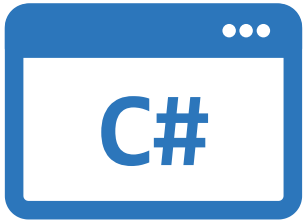
Application Insights for console applications - files

```
using System.IO;

// Reads ApplicationInsights.config file if present
TelemetryConfiguration config = TelemetryConfiguration.Active;

TelemetryConfiguration configuration =
TelemetryConfiguration.CreateFromConfiguration(File.ReadAllText("C:\\ApplicationInsight
s.config"));

var telemetryClient = new TelemetryClient(configuration);
```



Application Insights for console applications - code

```
var module = new DependencyTrackingTelemetryModule();

// Prevent Correlation Id to be sent to certain endpoints.
module.ExcludeComponentCorrelationHttpHeadersOnDomains.Add("core.windows.net");
// enable known dependency
module.IncludeDiagnosticSourceActivities.Add("Microsoft.Azure.EventHubs");

// initialize the module
module.Initialize(configuration);
```

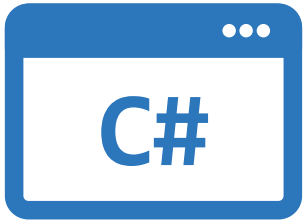


Application Insights for desktop apps

- Can be configured in a manner very similar to Application Insights for console apps
- Install latest **Microsoft.ApplicationInsights** NuGet package
- Install latest **Microsoft.ApplicationInsights.DependencyCollector** NuGet package
- Set the instrumentation key
  - By using the static **TelemetryConfiguration.Active.InstrumentationKey** property
  - By using the **ApplicationInsights.config** file

Application Insights for desktop apps - code

```
public partial class Form1 : Form
{
    private TelemetryClient tc = new TelemetryClient();
    private void Form1_Load(object sender, EventArgs e)
    {
        // Alternative to setting ikey in config file:
        tc.InstrumentationKey = "key copied from portal";
        // Set session data:
        tc.Context.User.Id = Environment.UserName;
        tc.Context.Session.Id = Guid.NewGuid().ToString();
        tc.Context.Device.OperatingSystem = Environment.OSVersion.ToString();
        // Log a page view:
        tc.TrackPageView("Form1");
        ...
    }
}
```



Application Insights platforms

- Official support – languages
  - .NET (C# & Microsoft Visual Basic)
  - Java
  - JavaScript
  - Node.js
- Unofficial support – languages
  - F#
  - PHP
  - Python
  - Ruby
- Official support –platforms/frameworks
  - ASP.NET (including ASP.NET Core)
  - Android
  - Angular
  - Azure (Azure App Service, Azure Cloud Services, Azure Functions)
  - Docker
  - Glimpse
  - iOS
  - Java 2 Platform Enterprise Edition (J2EE)
  - OS X
  - Spring
  - Universal Windows Platform (UWP)
  - Windows Communication Foundation (WCF)

Other monitoring tools

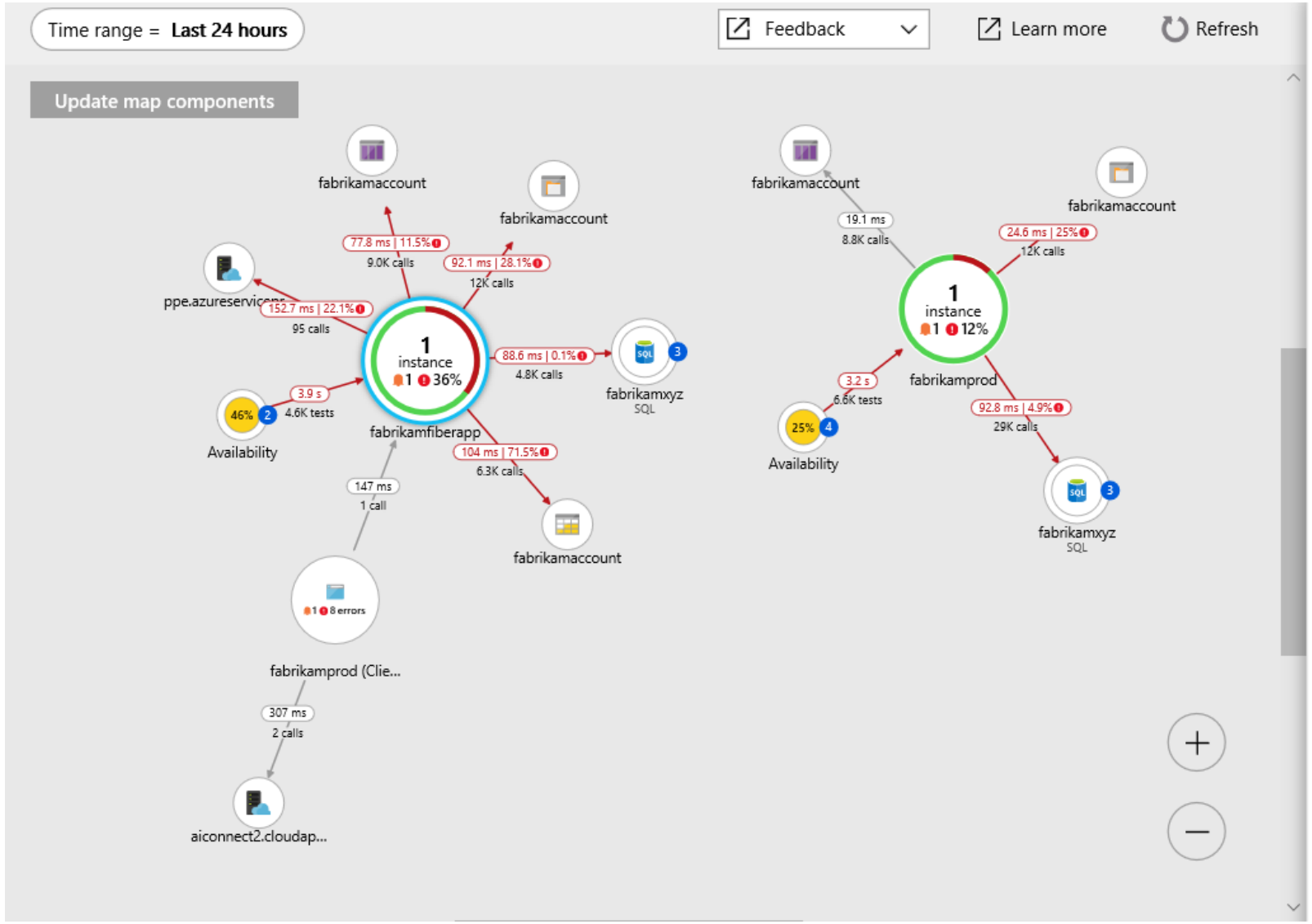
- Cloudyn
  - Manages and optimizes multi-platform, hybrid cloud deployments to help enterprises fully realize their cloud potential. The software as a service (SaaS) solution delivers visibility into usage, performance, and cost. It provides insights and actionable recommendations for smart optimization and cloud governance.
- AppDynamics
  - Application Performance Management (APM), which enables application owners to rapidly troubleshoot performance bottlenecks and optimize the performance of their applications running in an Azure environment.
- Datadog
  - Monitoring service that gathers monitoring data from your containers within your Azure Container Service cluster. Datadog has a Docker Integration Dashboard, where you can view specific metrics within your containers. Metrics gathered from your containers are organized by CPU, memory, network, and I/O.

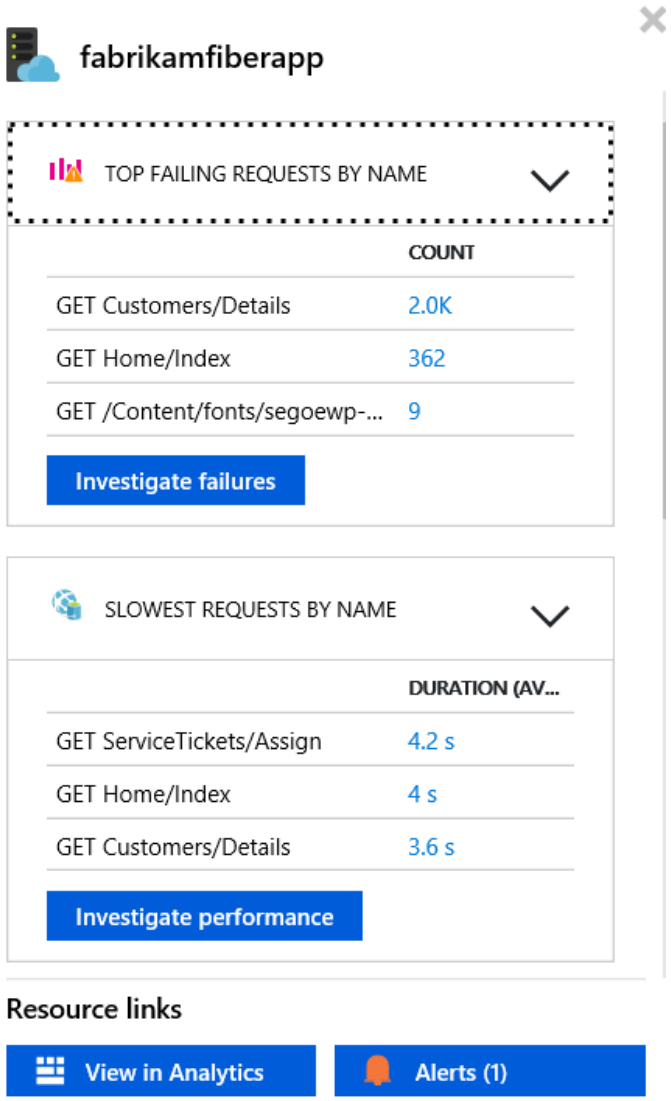
Other monitoring tools (continued)

- The Elasticsearch, Logstash, and Kibana (ELK) stack
  - Combination of Elasticsearch, Logstash, and Kibana that provides an end-to-end stack that can be used to monitor and analyze logs in your cluster. The ELK stack is popular for monitoring container clusters, because the monitoring stack itself is open source and already containerized.
- New Relic Application Performance Management
  - Popular APM add-in for .NET applications. New Relic Application Performance Management can be used similarly in the Azure platform and has first-class support in role-based access control scenarios.

Lesson 02: Analyze and troubleshoot solutions by using Azure Monitor

Application Map



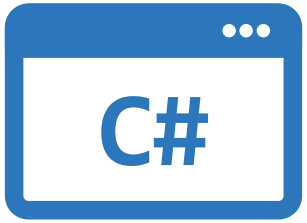


Components

- Components are independently deployable parts of your distributed/microservices application
  - Components are different from "observed" external dependencies
  - Components run on any number of server/role/container instances
- Components can be separate Application Insights instrumentation keys, or different roles reporting to a single Application Insights instrumentation key

Application Map - code

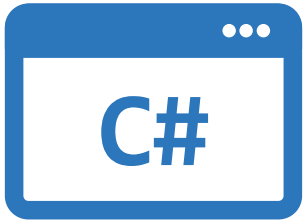
```
using Microsoft.ApplicationInsights.Channel;
using Microsoft.ApplicationInsights.Extensibility;

namespace CustomInitializer.Telemetry
{
    public class MyTelemetryInitializer : ITelemetryInitializer
    {
        public void Initialize(ITelemetry telemetry)
        {
            if (string.IsNullOrEmpty(telemetry.Context.Cloud.RoleName))
            {
                //set custom role name here
                telemetry.Context.Cloud.RoleName = "RoleName";
            }
        }
    }
}
```



Application Map - configuration

```
<ApplicationInsights>
    <TelemetryInitializers>
        <!-- Fully qualified type name, assembly name: -->
        <Add Type="CustomInitializer.Telemetry.MyTelemetryInitializer, CustomInitializer"/>
        ...
    </TelemetryInitializers>
</ApplicationInsights>
```

```
using Microsoft.ApplicationInsights.Extensibility;
using CustomInitializer.Telemetry;

protected void Application_Start()
{
    ...
    TelemetryConfiguration.Active.TelemetryInitializers.Add(
        new MyTelemetryInitializer()
    );
}
```



View activity logs to audit actions on resources

- Through activity logs, you can determine:
  - What operations were taken on the resources in your subscription
  - Who initiated the operation (although operations initiated by a back-end service do not return a user as the caller)
  - When the operation occurred
  - The status of the operation
  - The values of other properties that might help you research the operation

Auditing in Azure PowerShell

```
Get-AzLog -ResourceGroup ExampleGroup

Get-AzLog -ResourceGroup ExampleGroup -StartTime 2015-08-28T06:00 -EndTime 2015-09-10T06:00

Get-AzLog -ResourceGroup ExampleGroup -StartTime (Get-Date).AddDays(-14)

Get-AzLog -ResourceGroup ExampleGroup -StartTime (Get-Date).AddDays(-14) | Where-Object OperationName -eq Microsoft.Web/sites/stop/action

Get-AzLog -ResourceGroup deletedgroup -StartTime (Get-Date).AddDays(-14) -Caller someone@contoso.com

Get-AzLog -ResourceGroup ExampleGroup -Status Failed
```

Auditing in Azure PowerShell – retrieve specific operation

```
((Get-AzLog -Status Failed -ResourceGroup ExampleGroup -
DetailedOutput).Properties[1].Content["statusMessage"] | ConvertFrom-Json).error

code message
---- -------
DnsRecordInUse DNS record dns.westus.cloudapp.azure.com is already used by another
public IP.
```

Monitor availability and responsiveness of a website

- Use test to monitor availability and responsiveness
  - Tests send web requests to the application at regular intervals
  - Tests send requests from around the world
- Tests can point to any HTTP or HTTPs endpoint
  - Even if it's not hosted on Azure
- Two types of availability tests
  - URL ping test
    - Simple test that you can create in the Azure portal
  - Multi-step web test
    - Created in Visual Studio Enterprise and uploaded to the portal

Microsoft

Review

- Configure instrumentation in an app or service
- Analyze and troubleshoot solutions by using Azure Monitor