Application Insights

CSCI E-94

Fundamentals of Cloud Computing - Azure

Joseph Ficara

Agenda

- Overview
  - What is Application Insights?
  - Why do you care?
- Deployment and Configuration
- Customized Logging
- Snapshot Debugging

Overview
What is application insights

- Application Performance Management
  - A service for Developers and DevOps
- Provides for
  - Extensible logging
    - With full search and analytics capabilities
  - Failure detection
    - With analysis and enriched telemetry
  - Live monitoring
  - Dependency tracking and information
  - Detection of performance anomalies

Overview
Application Insights Provides for

- Diagnostic tools
  - See: https://bit.ly/3H9yedP
  - Multiple tech stacks including
    - C#|VB (.NET)
    - Java
    - JavaScript
    - Node.js
    - Python
  - Multiple platforms
    - Cloud (Any cloud!)
    - On-premises
    - Hybrid

Overview

What does Application Insights Monitor? …

- Request Rates
  - Response Times
  - Failure Rates
- Dependency rates
  - Response times
  - Failure rates

Overview

What does Application Insights Monitor? …

- Page views and load performance
- AJAX calls
- User and Session Counts
- Performance Counters
  - Windows & Linux
- Host Diagnostics
  - Docker & Azure

Overview

What does Application Insights Monitor? …

- Correlated diagnostic trace logs
  - From request into your tech stack
  - End to End correlation
- Custom Events
  - Business events
    - Account creation
    - Assessments taken

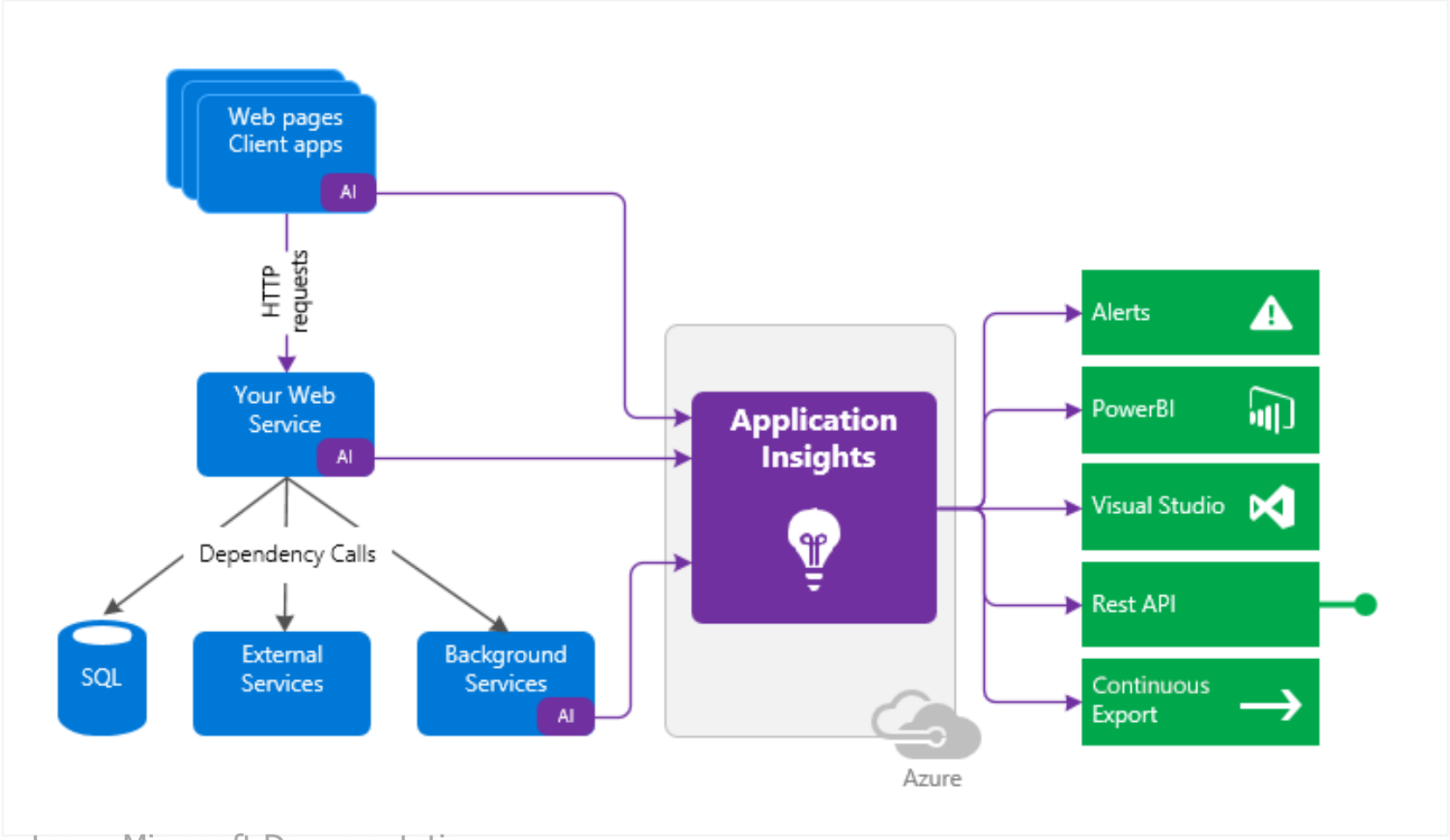
Overview

What does Application Insights Monitor? …

- Custom Metrics
  - Internal Ids and User Context
  - Request payload
  - Cosmos DB query costs

Overview

How does Application Insights Work?



Courtesy - Microsoft Documentation:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview

Overview
Why do you care?

- Save time & money
- Proactive instead of reactive
  - Identify issues before the happen
  - Optimize
  - Spend time where it matters
- Root caused analysis
  - Efficiently

Overview

Deployment and Configuration

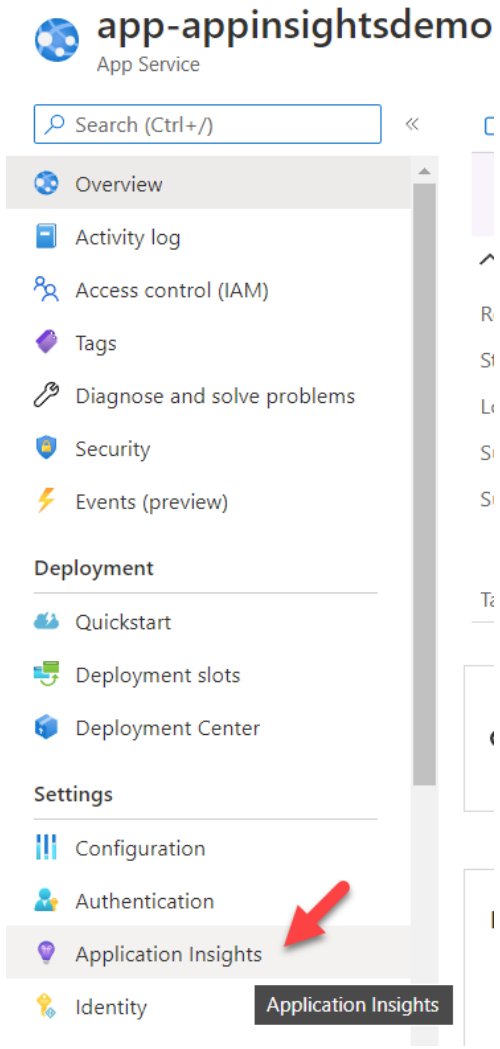
- Two "levels" of Application Insights
  - Portal configured
    - No change to code necessary
  - Application Insights SDK
    - Enrich telemetry
    - Add your own custom
      - Events
      - Trace log data
      - Metrics

Deployment and Configuration

- Portal Configured requires 5 steps
  1. Go to your app service
  2. Click Application Insights
  3. Turn on Application Insights
  4. Create a new Application Insights resource
  5. Instrument your application
     - Accomplished without code changes!

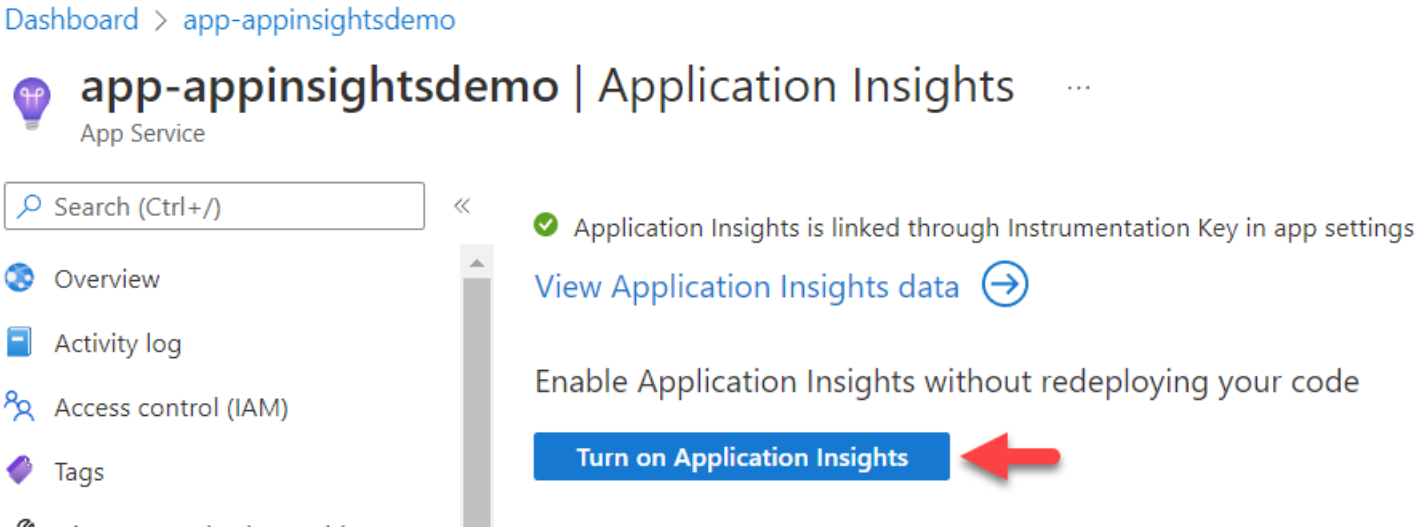
Deployment and Configuration

2 Click Application Insights



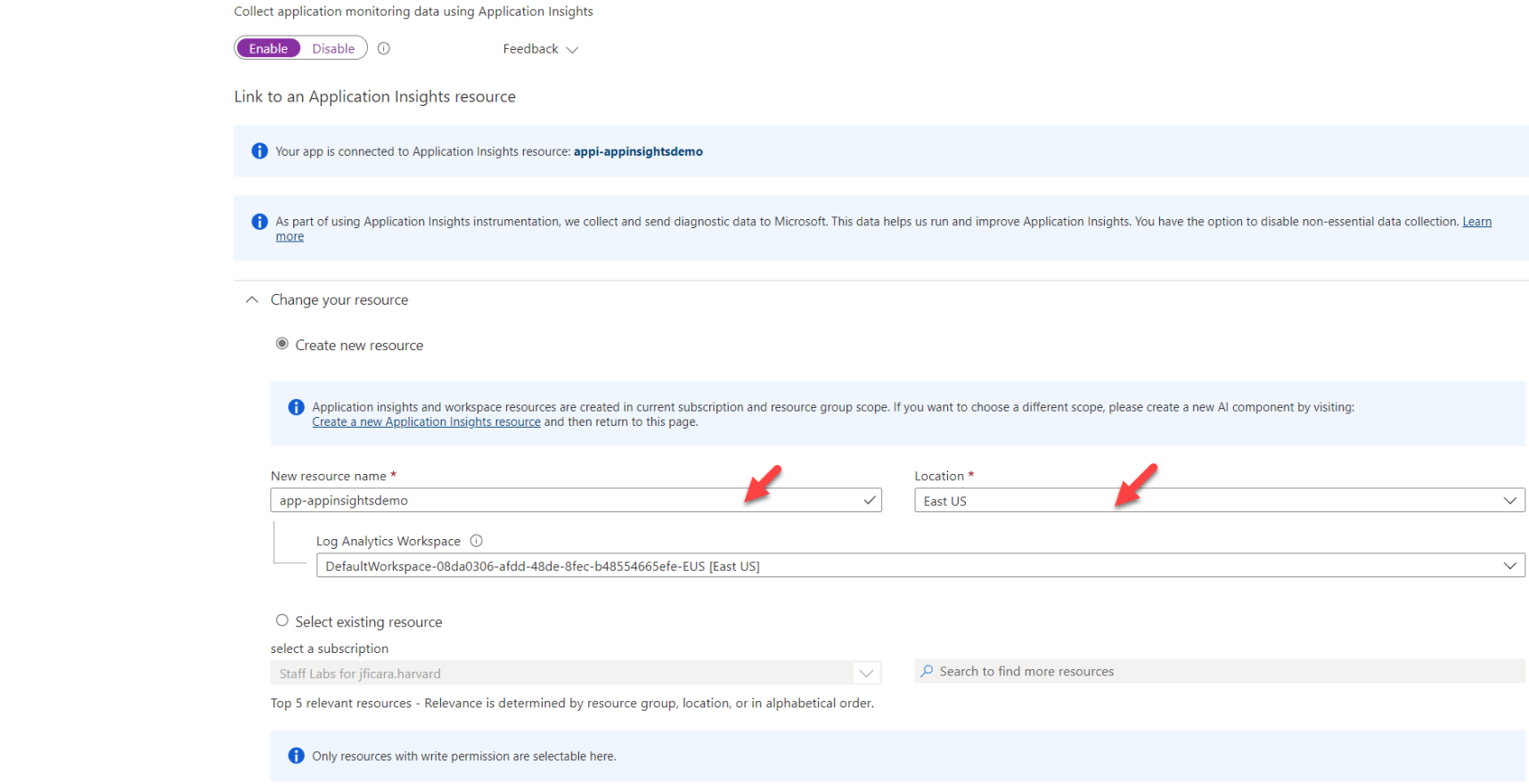
Deployment and Configuration

3 Turn on Application Insights



Deployment and Configuration

4 Create a new Application Insights resource



Deployment and Configuration

5 Instrument your application

Instrument your application

Info .NET .NET Core Node.js Java Python

Collection level

Gain full APM visibility with correlation across boundaries, improved accuracy, and rich usage analytics.

How to get the most out of your APM data collection

Recommended Disabled

Interop with Application Insights SDK (preview)

Load the extension side-by-side with SDK and use it to send telemetry (disables the Application Insights SDK)

On Off

SDK required for Linux

Profiler

Collect profiling traces that help you see where time is spent in code.

How to use Profiler to identify code that slowed down your web app

On Off

Profiler requires basic or above

Snapshot debugger

Collect call stacks for your application when an exception is thrown.

How to use Snapshot Debugger

On Off

Show local variables for your application when an exception is thrown.

On Off

SQL Commands

Enable dependency tracking to show compiled SQL command and local variables in a web request.

On Off

Apply

Demo

Portal Configured Application Insights

Azure Portal

app-appinsightssimpledemo-cscie94-win
appi-appinsightssimpledemo-cscie94-win

Customized Logging

- Why do you care?
  - You can write your own
    - Log entries
    - Events
  - You can enhance exceptions
    - Add custom metadata

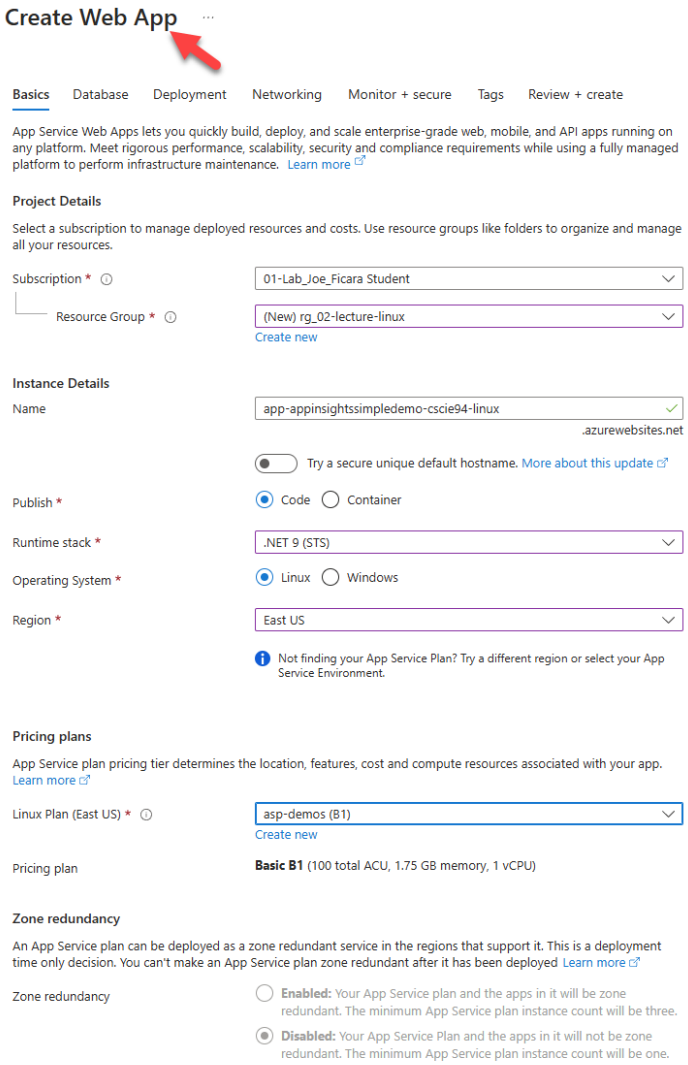
Customized Logging

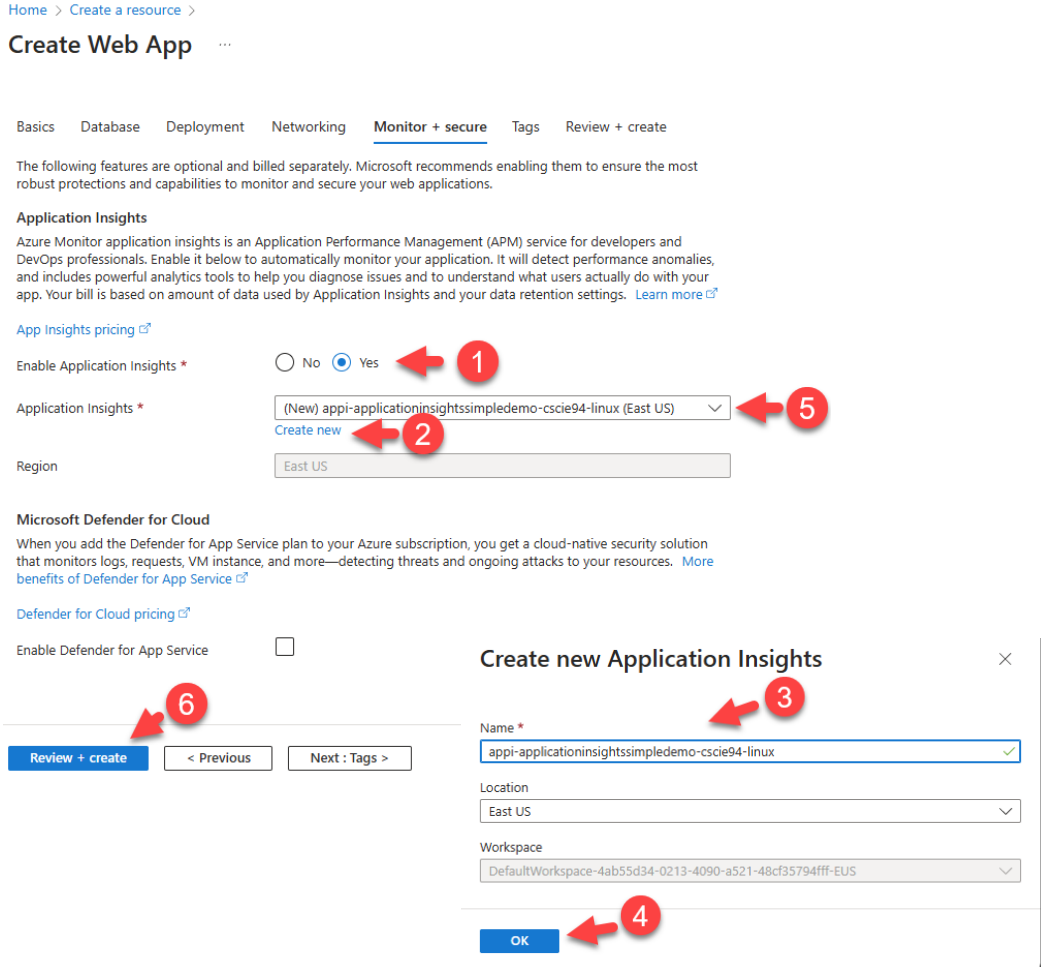
- How do you do it?
  - Utilize the **`TelemetryClient`** instance
    - This is not automatically provided
    - When adding Application insights

```
builder.Services.AddApplicationInsightsTelemetry(aiOptions);
```

Customized Logging

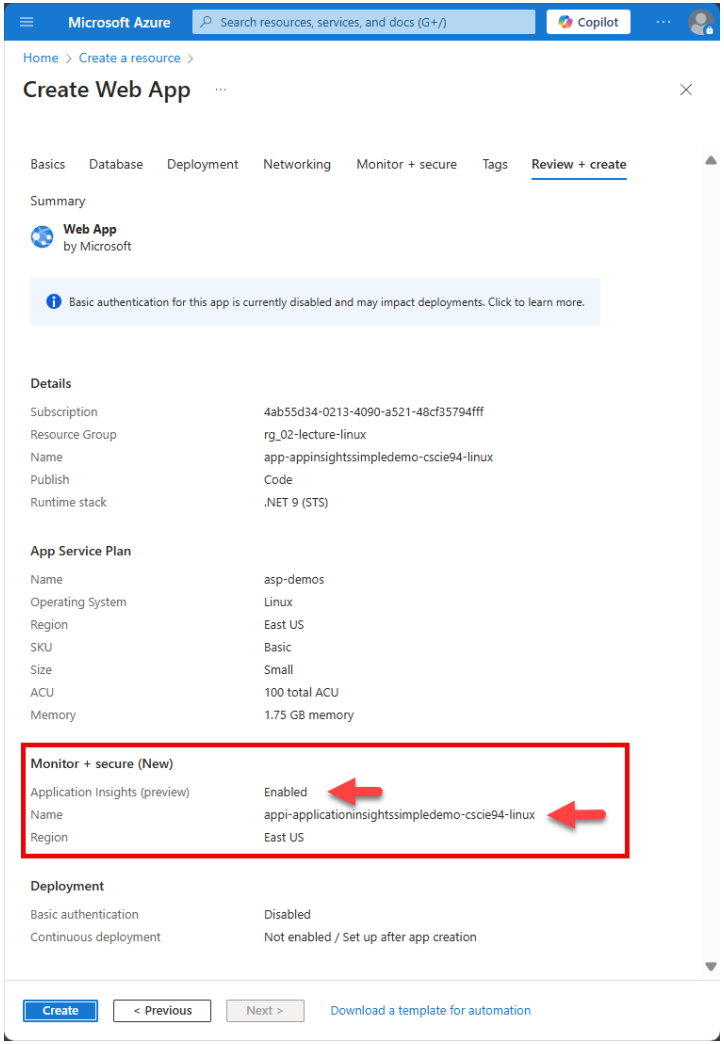
How do you do it? …





Customized Logging

How do you do it? …



Customized Logging
How do you do it? …

- Use the **`TelemetryClient`**
  - Log via **`TrackTrace`**
  - Enrich Exceptions via **`TrackException`**
  - Write Events via **`TrackEvent`**
  - Track Metrics via **`GetMetric(x).TrackValue`**
    - Where **`X`**
      - is the **`string`** name of the metric
      - Or **`MetricIdentifier`** instance
        - Supports namespace and metric id

ILogger
How do you do it?

- ILogger support is now built in but…
  - Requires additional settings
  - A sub object of Logging called
    - **`ApplicationInsights`**
  - Default behavior only logs
    - Warning
    - Error
    - Critical

Application Insights - ILogger

Settings

```
"Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.AspNetCore": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    },
    "ApplicationInsights": {
      "LogLevel": {
        "Default": "Information"
      }
    }
  }
```

Configuring Settings
In Azure

- Notes: http://bit.ly/38UjpdR
  - To set a setting in the Azure Portal
    - Setting name is combination of
      - Section name and property name
    - Windows
      - `<section name>:<property name>`
        - Ex: `CustomerLimits:MaxCustomers`
    - Linux
      `<section name>__<property name>`
      - Ex: `CustomerLimits__MaxCustomers`
        - The __ is a double under score

Demo

Customized Logging

AppInsightsDemoSolution.sln

CustomersController.Rest.cs

Line 92–100 (Setup metrics and custom properties)

Line 150-163 (Custom event with metrics)

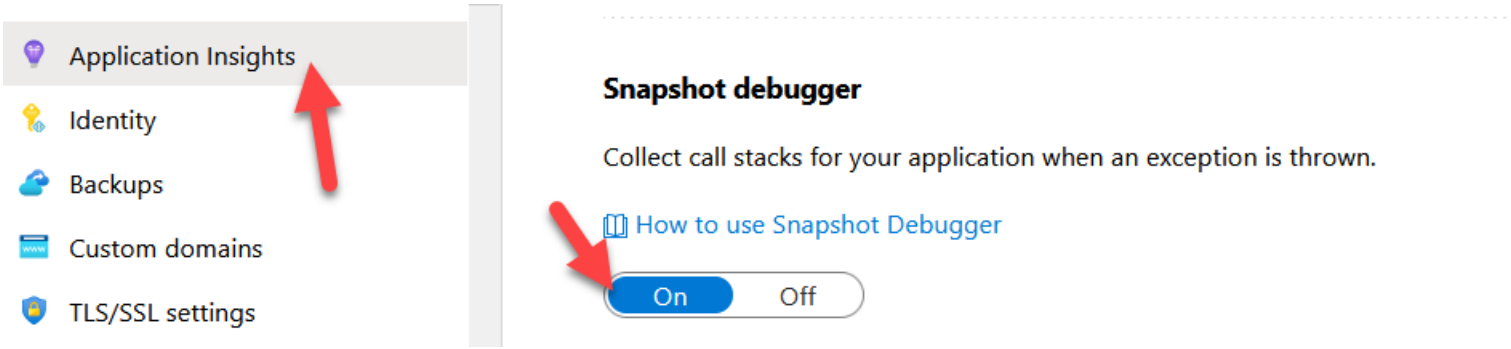
Line 182 (Enrich an exception)

Snapshot Debugging

- What is it?
  - Ability to capture snapshots of your code
    - Local variables
    - Method parameters
    - Stack trace
      - Multiple threads
- Why do you care?
  - Helps diagnosing production issue
    - Faster & Easier

Snapshot Debugging

- How do you do it:
  - Add a nuget package
    - `Microsoft.ApplicationInsights.SnapshotCollector`
  - Turn on snapshot debugging in portal
    - Go to app service and click **Application Insights**



Snapshot Debugging

- Enable snapshot collection in code
  - Configure method in **`program.cs`** file
  - builder.Services.AddSnapshotCollector
    - Bind to settings
      - SnapshotCollectorConfiguration
    - Settings definitions
      - See: http://bit.ly/3uRbARK

Snapshot Settings

SnapshotCollectorConfiguration in appsettings.json

```
...
  // DEMO: Setup snapshot debugging
  builder.Services.AddSnapshotCollector((configuration) =>
       Configuration.Bind(nameof(SnapshotCollectorConfiguration),
       configuration));
```

Snapshot Settings

SnapshotCollectorConfiguration in appsettings.json

```
"SnapshotCollectorConfiguration": {
   "IsEnabledInDeveloperMode": false,
   "ThresholdForSnapshotting": 1,
   "MaximumSnapshotsRequired": 3,
   "MaximumCollectionPlanSize": 50,
   "ReconnectInterval": "00:15:00",
   "ProblemCounterResetInterval": "1.00:00:00",
   "SnapshotsPerTenMinutesLimit": 1,
   "SnapshotsPerDayLimit": 30,
   "SnapshotInLowPriorityThread": true,
   "ProvideAnonymousTelemetry": false,
   "FailedRequestLimit": 10
 }
```

Snapshot Settings

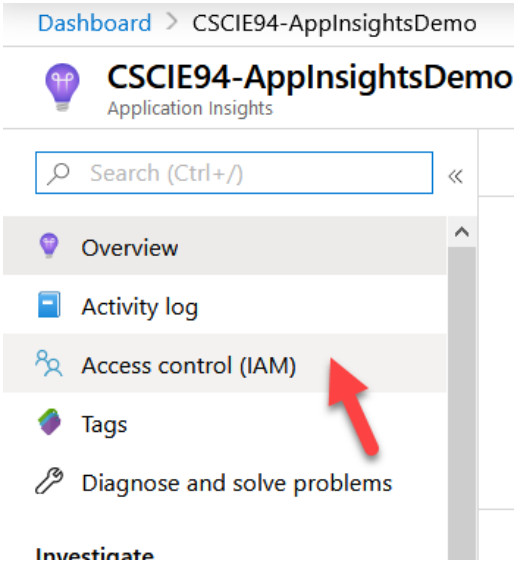
- ThresholdForSnapshotting = 1
  - Snapshots are taken on **second occurrence**
- IsEnabledInDeveloperMode = false
  - Not collected
    - **When debugging in Visual Studio**
- See: http://bit.ly/3uRbARK

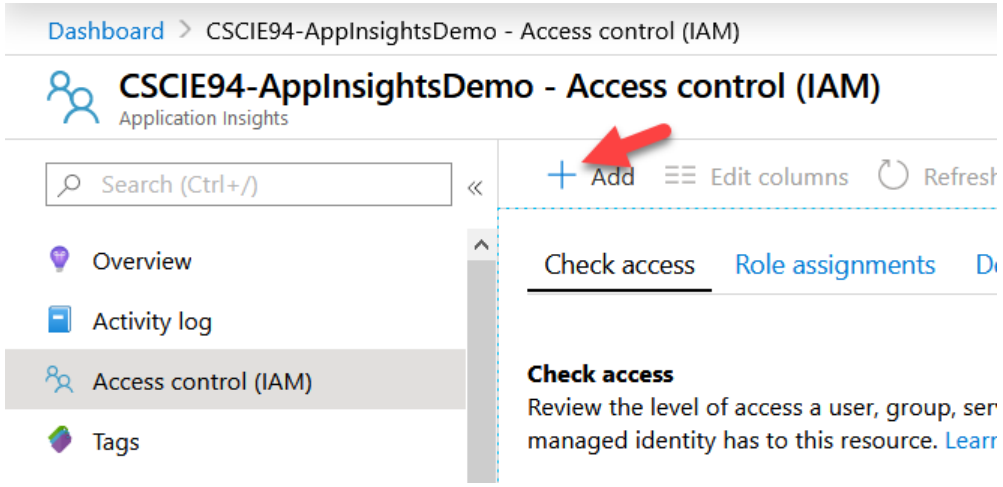
Snapshot Debugging

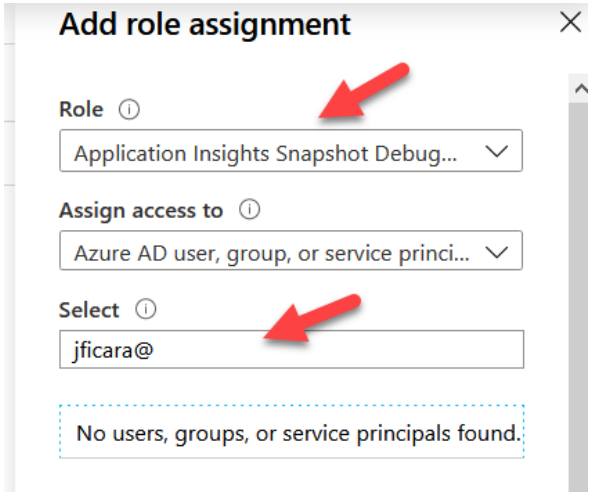
- Two options
  - Grant yourself access
    - Go to your Application Insights Instance
      - Can also apply to resource group or subscription
    - Click on Access control (IAM)
    - Click on Role assignments
    - Add yourself to
      - Application Insights Snapshot Debugger
  - Automatic workflow
    - When attempting to look at snapshot details

Snapshot Debugging

Grant yourself access

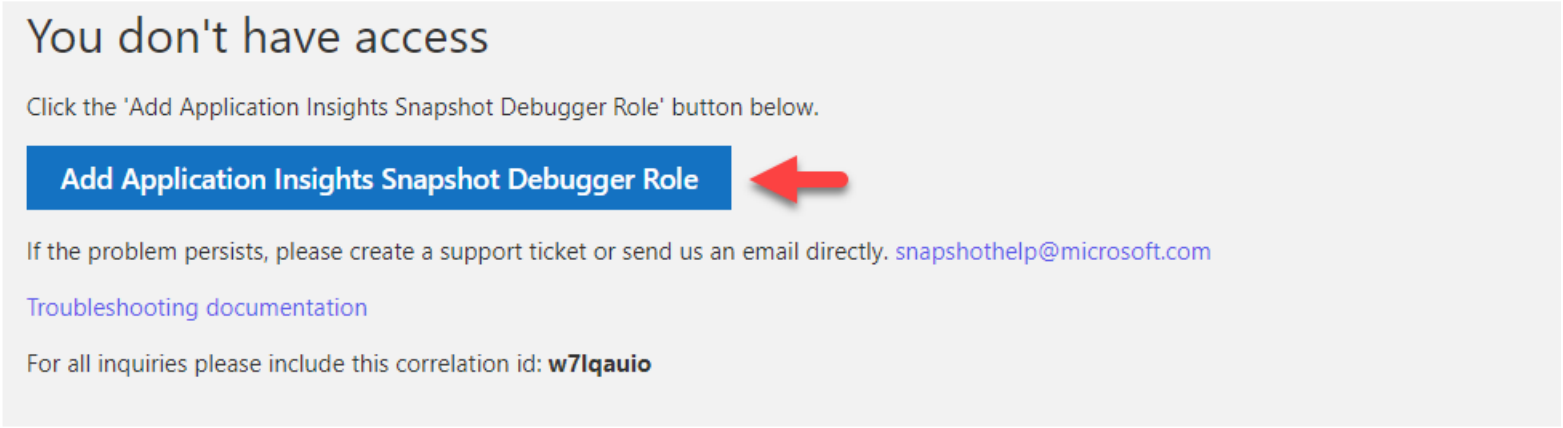






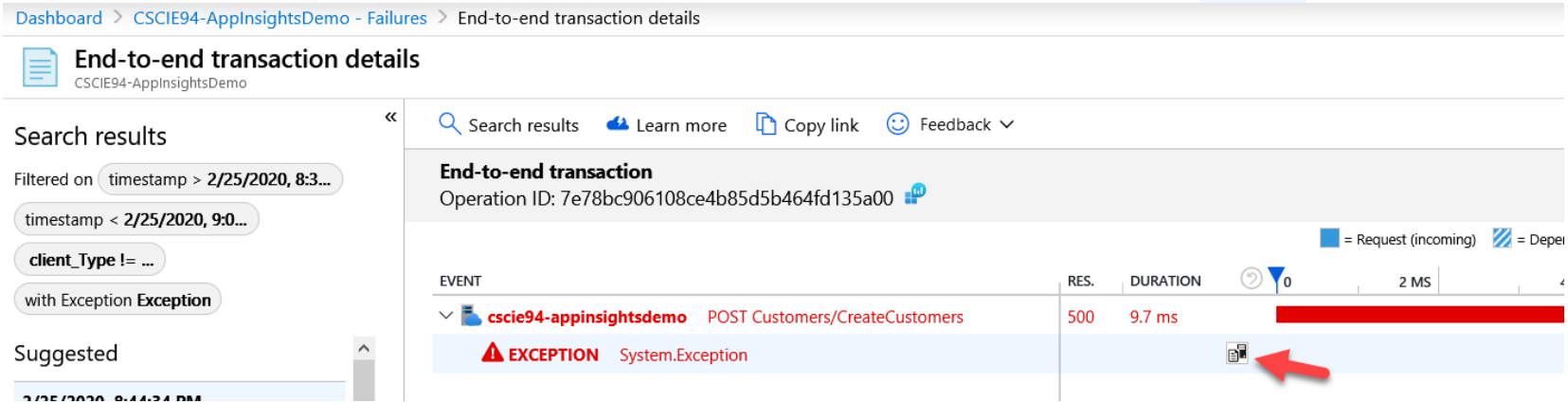
Snapshot Debugging

Automatic Role Addition



Snapshot Debugging

- Where are snapshot debug images?
  - Failures -> Exceptions
    - Click the Count look for the Icon 



Snapshot Debugging

- What can you do with it?
  - Look at the information in the portal
  - Download the snapshot
    - Open it in **Visual Studio Enterprise**
    - Use the familiar debugging experience
      - Look at stack traces
      - Variables
      - Etc…

Snapshot Debugging

- Supported Environments
  - Azure App Service
  - Azure Functions
  - Azure Cloud Services
    - Running OS family 4 or later
  - Running Windows Server 2012 R2 or later
    - Azure Service Fabric
    - Azure Virtual Machines
    - On-premises virtual or physical machines
      - Or Windows 8.1 or later

Snapshot Debugging

- Limitations
  - Not supported on Linux
  - Interop combined with SDK
    - Looses telemetry and can crash on startup
    - Use codeless snapshot if using interop
    - Use SDK Only
  - Consumption Free and Shared Tiers
    - Not supported
  - You need Visual Studio Enterprise
    - To view downloaded **diagsession** files

Demo

Snapshot Debugging

`AppInsightsDemoSolution.sln`

Snapshot Debugging

Links & Resources

- Application Insights Overview
  - http://bit.ly/2SZapib
- Telemetry Client
  - http://bit.ly/2wNy38M
- Custom Events & Metrics
  - http://bit.ly/2SWVNzV
- Troubleshooting Snapshot Debugging
  - http://bit.ly/2VnVNe0
- Live Monitoring Authentication using AzureAD
  - http://bit.ly/3HrgsW7

Links & Resources

- Configuration Settings
  - http://bit.ly/3uRbARK
- Custom Metrics
  - http://bit.ly/2wLbj9j
- Features
  - http://bit.ly/3pphWUA
  - http://bit.ly/37ZQBzF