Implementing Centralized Logging in Microservices



Steve Gordon
MICROSOFT DEVELOPER TECHNOLOGIES MVP
@stevejgordon www.stevejgordon.co.uk



Overview



Learn to centralize logs using the Elastic Stack (ELK)

Implement Serilog

- Send logs to Elasticsearch

Learn how to query log data in Kibana

Focus on coordinated logging with REST APIs

Learn to enrich logs with structured data

Learn about correlating logs over gRPC and Azure Service Bus



Centralizing Logs with the ELK Stack



Microservices Application





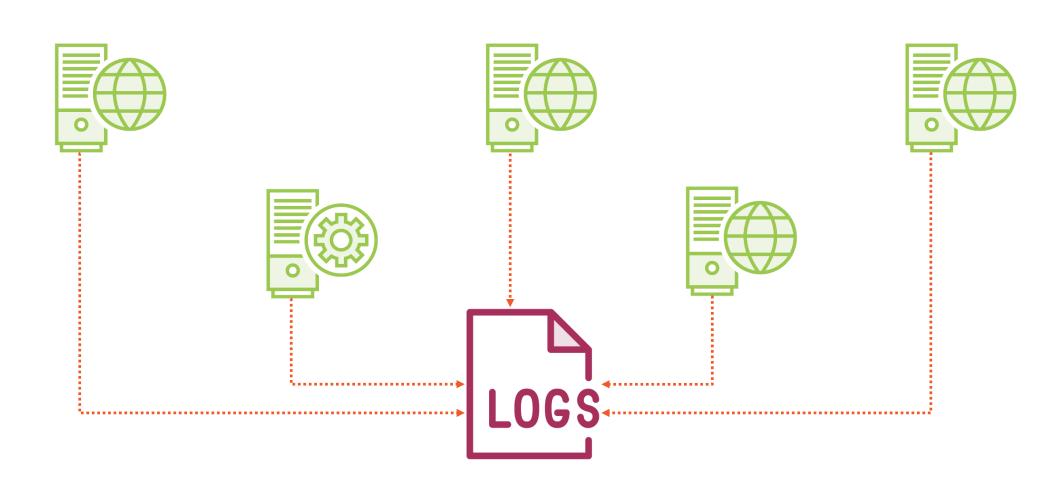








Microservices Application





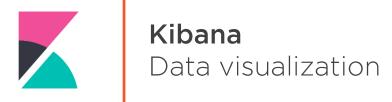
Elastic (ELK) Stack

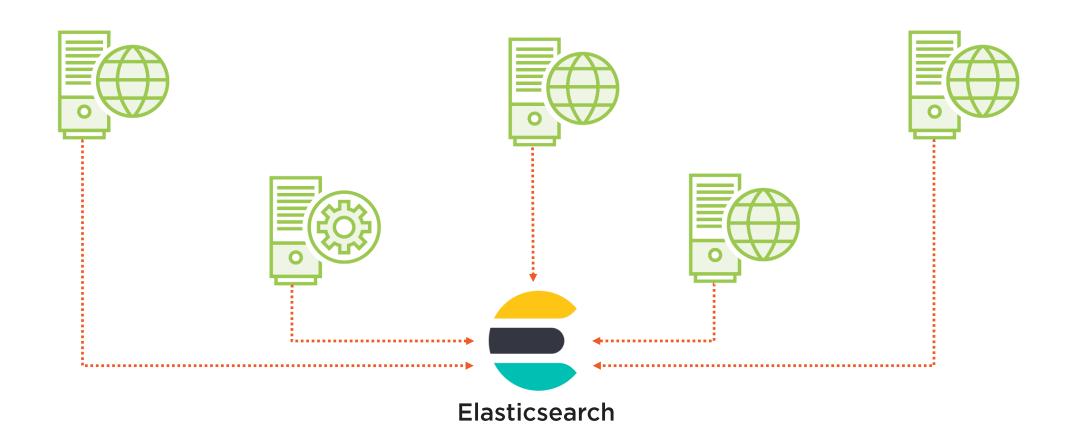


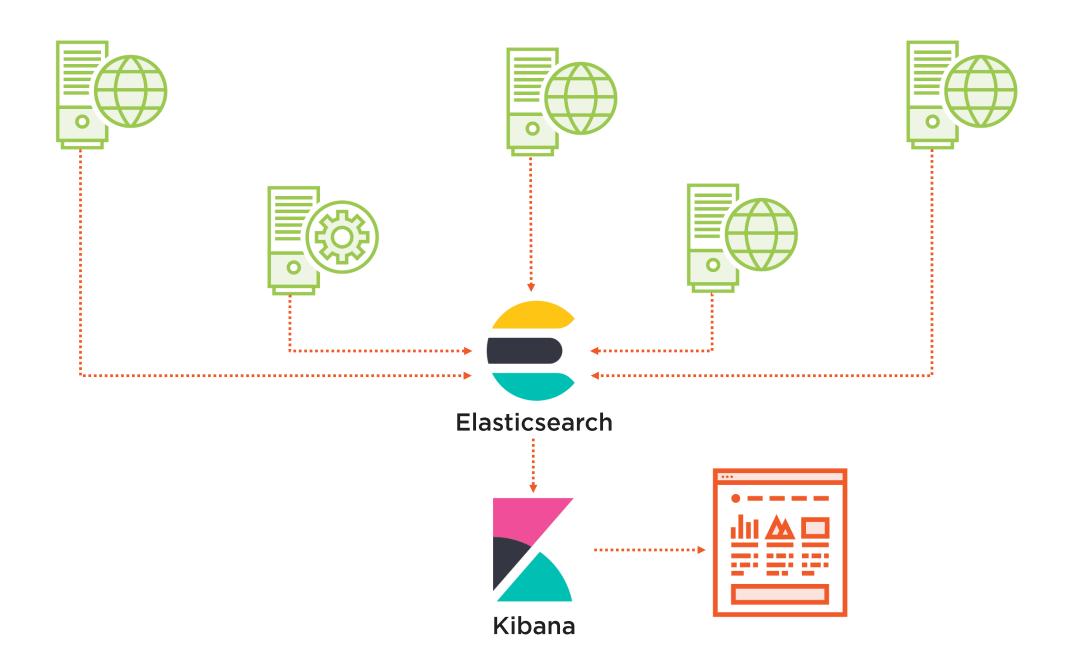
Elasticsearch
Schema-free document store



Logstash
Log shipping and processing











Install and configure Serilog

- Create a reusable extension method for configuration
- Update GloboTicket.Web to enable Serilog logging





Sinks

Serilog provides sinks for writing log events to storage in various formats.





Configure Serilog Elasticsearch sink





Run the application to write logs to Elasticsearch

Use Kibana to visualize log messages





Correlating logging between microservices

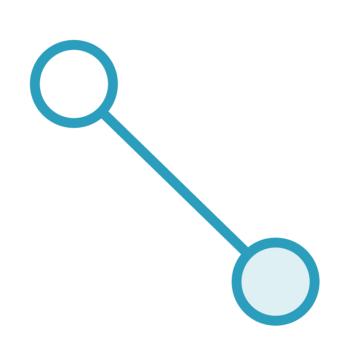
- Using the Trace ID



.NET Core Tracing



.NET Core Activities



Activity type provides distributed tracing

ASP.NET Core automatically starts an activity per request

- Includes a unique Trace (correlation) ID

HttpClient automatically forwards activity details

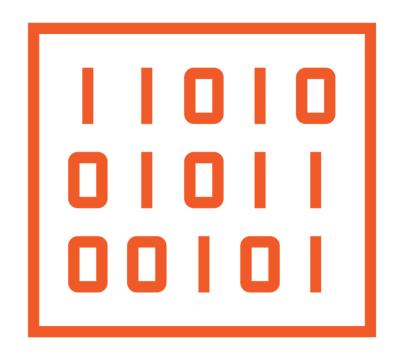
Includes a traceparent header on requests

When the header is included in a request, ASP.NET Core continues the trace



Controlling the Traceld Format





ASP.NET Core 3.1 defaults to bespoke hierarchical trace identifier format

Open standards are being created for application observability

- W3C distributed tracing

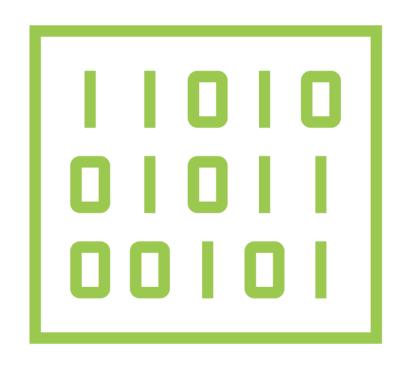
.NET Activity library supports using W3C trace identifiers



Introducing Structured Logging



Structured Log Data



Microsoft logging library supports structured log data

Attach complimentary information to logs

Serilog supports writing structured data to Elasticsearch

Additional fields can be used to filter, aggregate and sort logs



Adding Structured Log Data

LoggerExtensions.cs

```
logger.LogDebug("Received a success response from {Url}",
response.RequestMessage.RequestUri);
```

```
"Url": "https://localhost:5001/api/events/adc42c09-08c1-4d2c-9f96-2d15bb1af299"
}
```

Enrich logs with structured data to support searching and filtering.





Update Serilog to enrich log messages
Refresh the index pattern in Kibana





Improve exception log messages



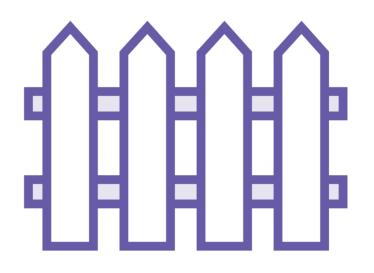


Introduce logging scopes

Learn how to supply common log data



Logging Scopes



Group a set of logical operations

Attach common data to each log created inside the scope

ASP.NET Core creates a scope per request

A scope is an IDisposable type which lasts until it is disposed

Serilog includes it own syntax for adding data to the logging context



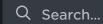


Correlate logs when using gRPC













Expand All









AUTHOR TOOLS

⚠ Home

Analytics

Author's nest

Author kit

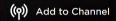
Microservices Communication in ASP.NET Core

by Gill Cleeren

This course will teach you how to architect a microservice-based application and show how to organize the communication between different microservices as well as the different front-ends.







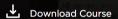


Table of contents

Description

Transcript

Exercise files

Discussion

Related Courses



Course author



Gill Cleeren

Gill Cleeren is a Microsoft Regional Director, MVP and Pluralsight author. Gill is a freelance solution architect living in Belgium. He focuses on web and mobile development and loves Xamarin. He's...

Course info

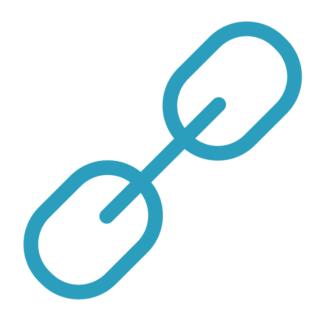
Level	Intermediate
Rating	**** (12)
My rating	****
Duration	3h 15m
Released	30 Sep 2020

Share course





gRPC Tracing



gRPC client library uses HttpClient with additional diagnostic events

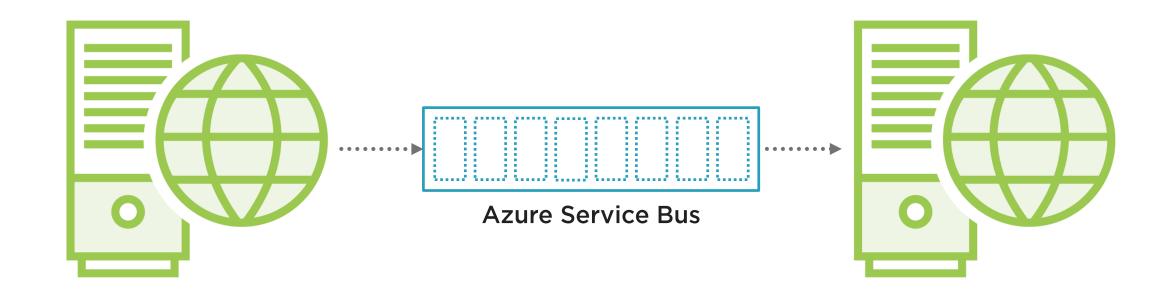
gRPC server library hosted on ASP.NET Core records events with gRPC specific metadata





Correlate logs when using Azure Service Bus









Other choices exist for correlation



Summary



Centralized logs using Serilog and the ELK stack

Used Kibana to view and filter log messages from all microservices

Correlated logs using the Trace ID

Enriched logs using structured data and logging scopes

Implemented log correlation over gRPC and an Azure Service Bus



Up Next:

Implementing Health Checks

