Microsoft

# Microsoft Java Developer Conference 2024

Code. Cloud. Community.

## Empowering Quarkus Applications with AI and OpenTelemetry on Kubernetes

## Who We Are!

Daniel Oh

Developer Advocate, Java Champion

X danieloh30

Brian Benz

Microsoft

Cloud Advocate, Java Champion

X bbenz

## Part 1: Understanding OpenTelemetry and Quarkus

## DEVELOPER

What is the **health** of my application?

What is the **root cause of errors and defects**?

What are the **performance bottlenecks** that could impact customer experience?

## Observability Pillars

### Metrics

Numbers describing a particular process or activity measured over intervals of time

### Logs

Immutable record of discrete events that happen over time

### Traces

Data that shows which line of coding is falling to gain better visibility at the individual user level for events that have occurred

# OpenTelemetry Components

**Specification**

Cross-language requirements and expectations for all implementations

**API** **SDK** **Data**

**Instrumentation**

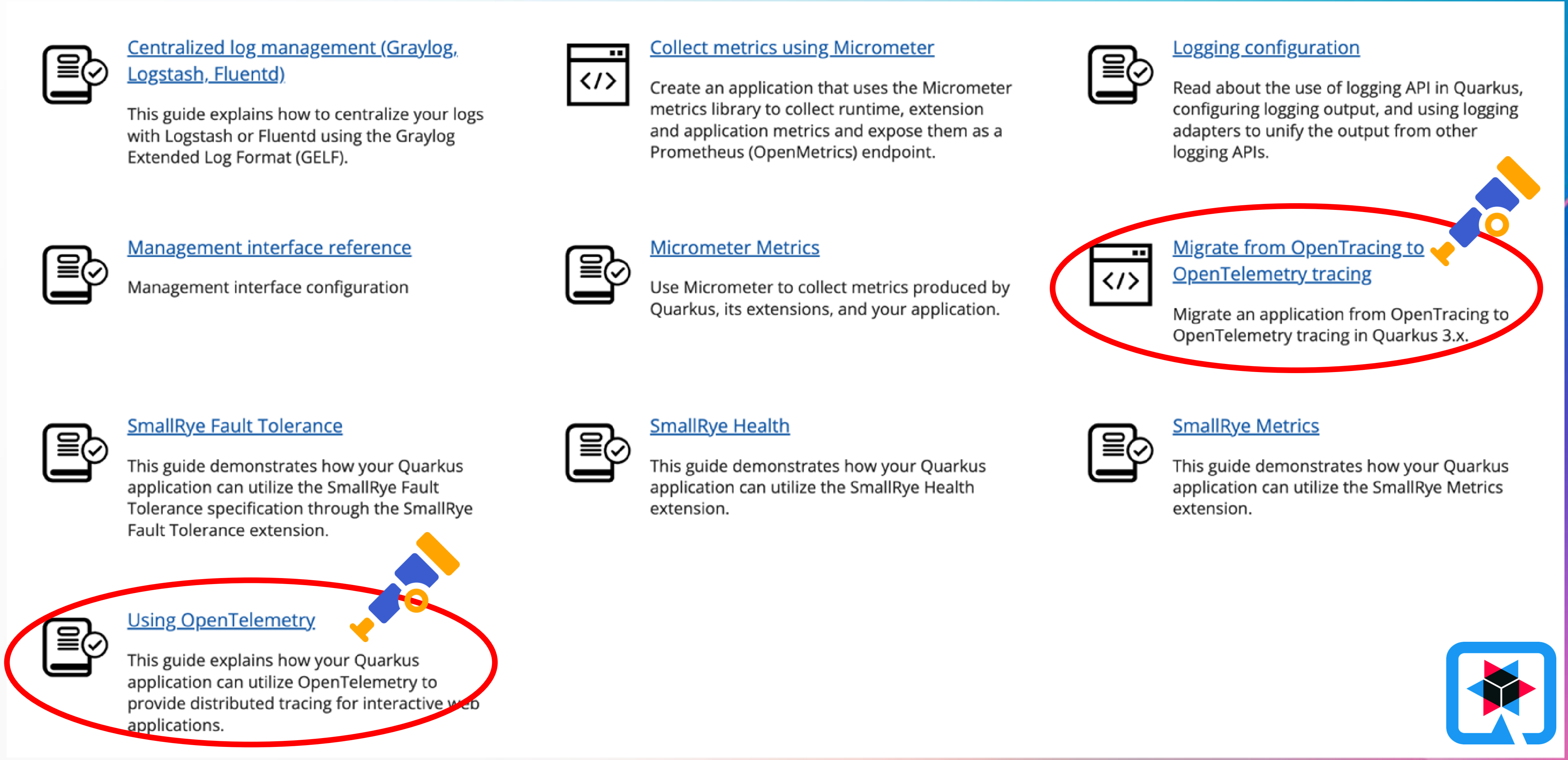
Make every library and application observable out-of-the-box

**Collector**

Vendor-agnostic implementation on how to receive, process and export telemetry data

# Observability in Quarkus

Container First | Imperative Reactive | Kube Native | Developer Joy | Community Standard



Centralized log management (Graylog, Logstash, Fluentd)

This guide explains how to centralize your logs with Logstash or Fluentd using the Graylog Extended Log Format (GELF).

Collect metrics using Micrometer

Create an application that uses the Micrometer metrics library to collect runtime, extension and application metrics and expose them as a Prometheus (OpenMetrics) endpoint.

Logging configuration

Read about the use of logging API in Quarkus, configuring logging output, and using logging adapters to unify the output from other logging APIs.

Management interface reference

Management interface configuration

Micrometer Metrics

Use Micrometer to collect metrics produced by Quarkus, its extensions, and your application.

Migrate from OpenTracing to OpenTelemetry tracing

Migrate an application from OpenTracing to OpenTelemetry tracing in Quarkus 3.x.

SmallRye Fault Tolerance

This guide demonstrates how your Quarkus application can utilize the SmallRye Fault Tolerance specification through the SmallRye Fault Tolerance extension.

SmallRye Health

This guide demonstrates how your Quarkus application can utilize the SmallRye Health extension.

SmallRye Metrics

This guide demonstrates how your Quarkus application can utilize the SmallRye Metrics extension.

Using OpenTelemetry

This guide explains how your Quarkus application can utilize OpenTelemetry to provide distributed tracing for interactive web applications.

## Part 2: Leveraging Azure AKS for Deployment and Management

# How Managed Kubernetes on Azure works

Automated upgrades, patches

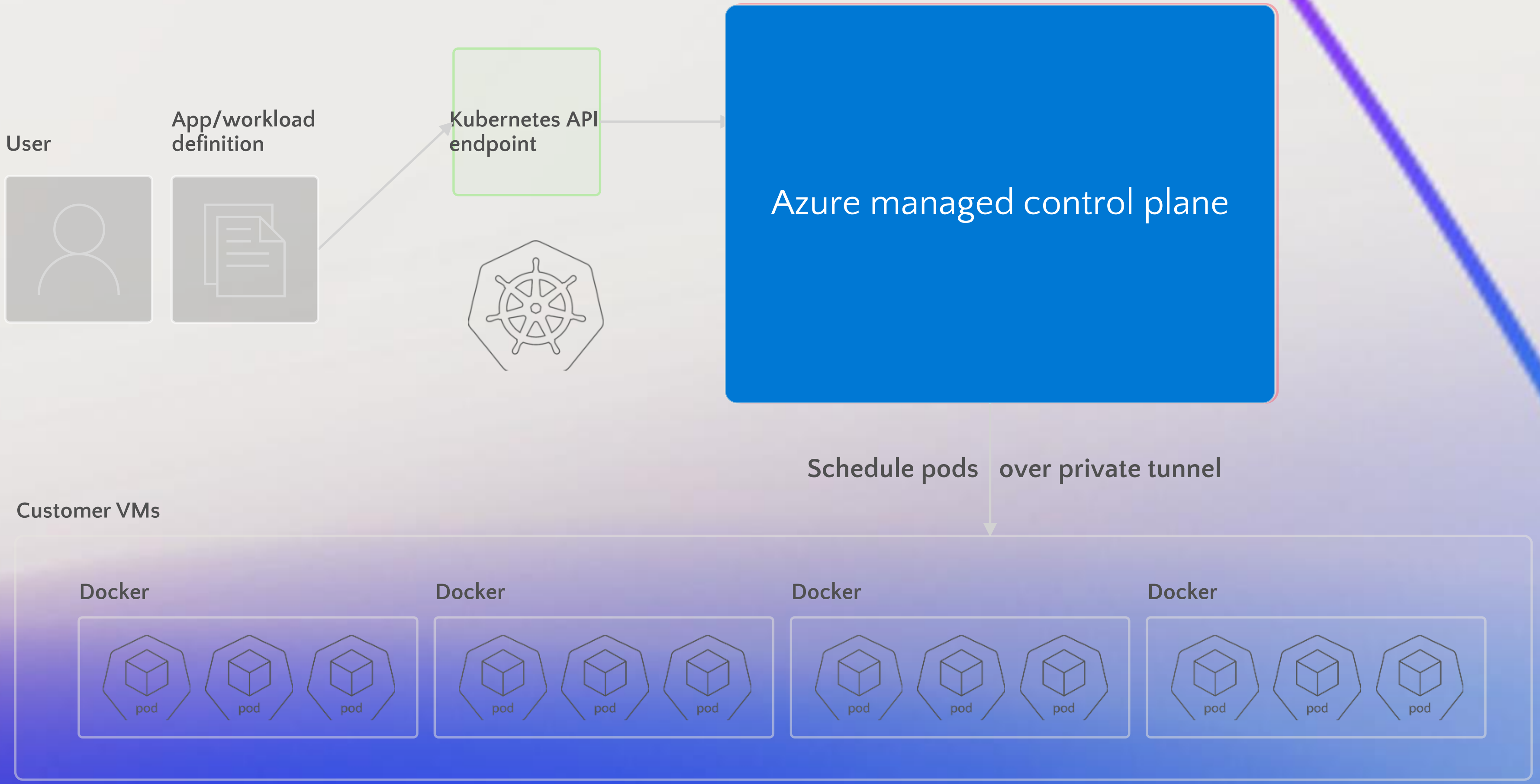
High reliability and availability

Easy and secure cluster scaling

Self-healing

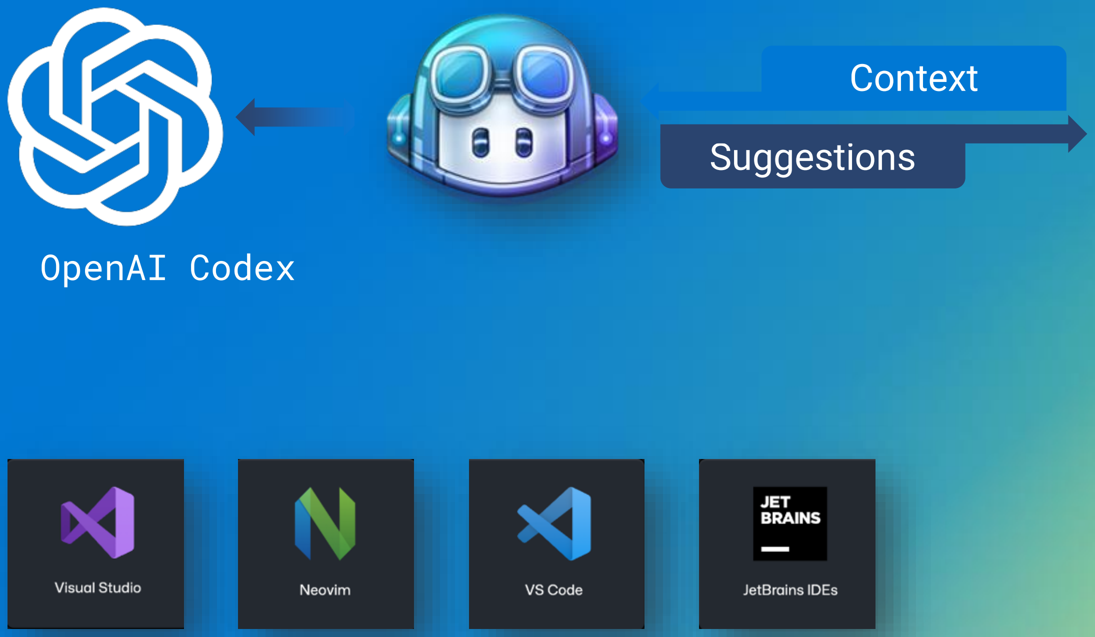
API server monitoring

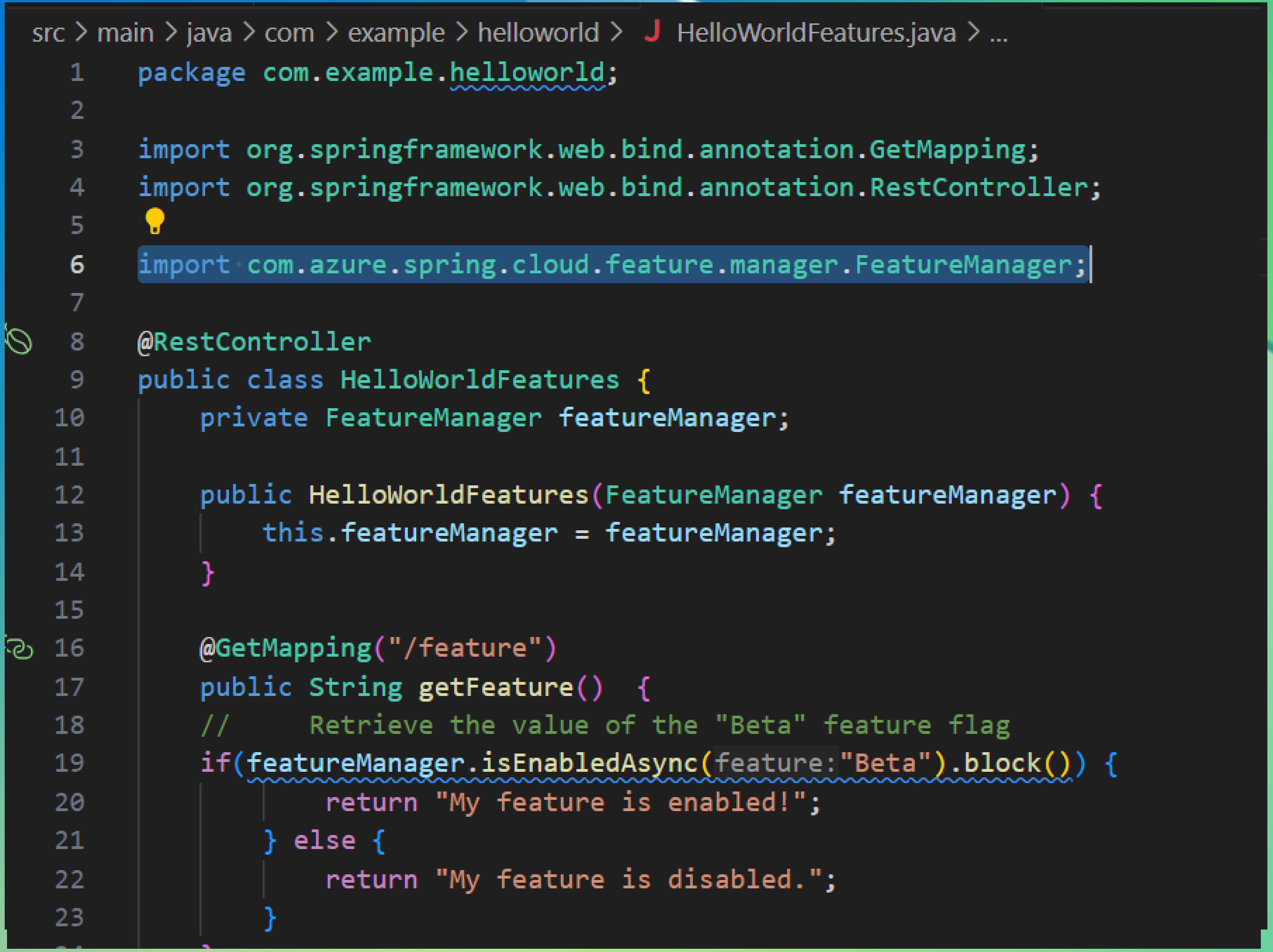
Control plane at no charge



## Part 3: Streamlining Telemetry with GitHub Copilot

# GitHub Copilot





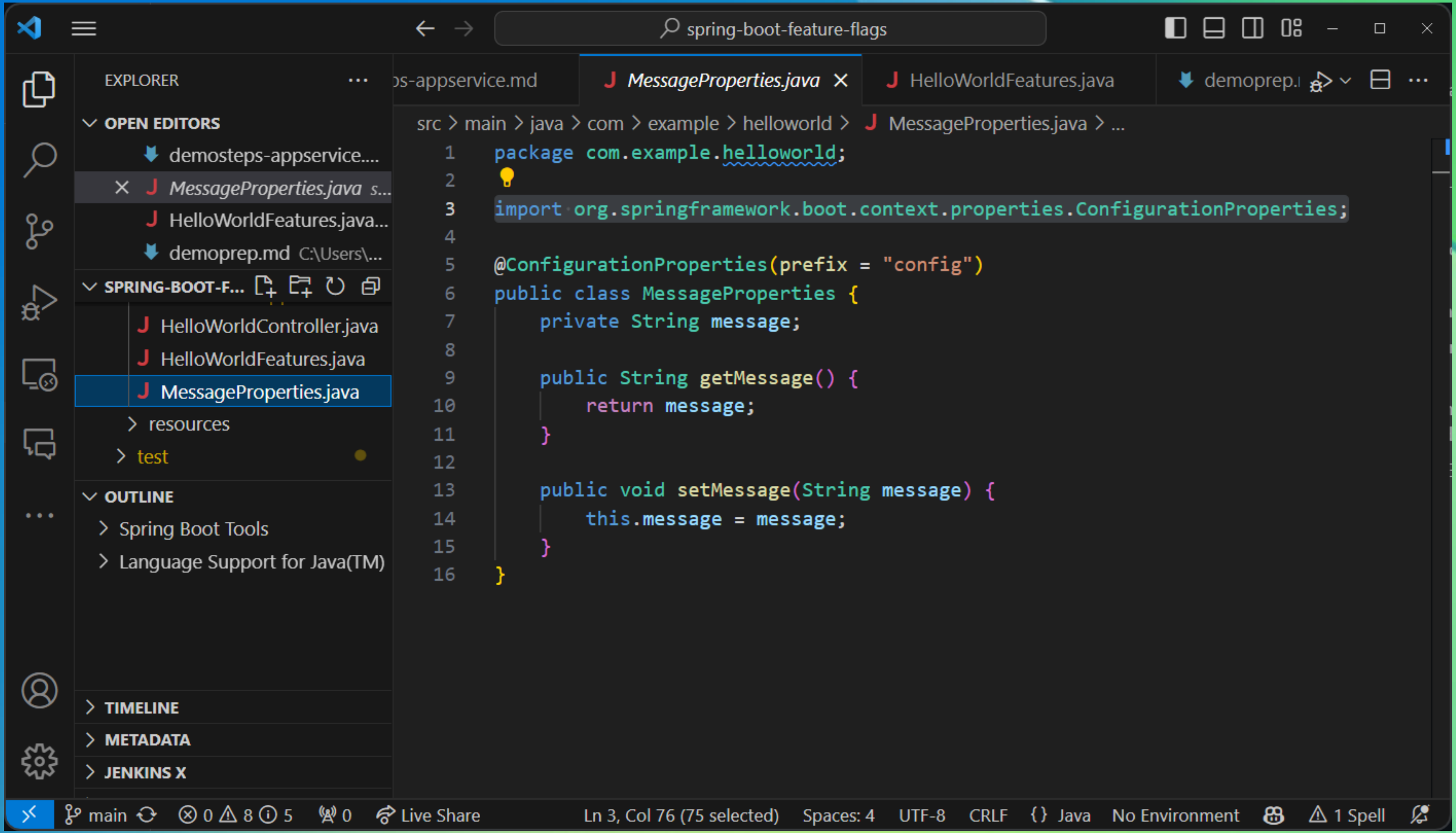
src > main > java > com > example > helloworld > HelloWorldFeatures.java > ...

```java
package com.example.helloworld;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import com.azure.spring.cloud.feature.manager.FeatureManager;

@RestController
public class HelloWorldFeatures {
    private FeatureManager featureManager;

    public HelloWorldFeatures(FeatureManager featureManager) {
        this.featureManager = featureManager;
    }

    @GetMapping("/feature")
    public String getFeature()  {
    //      Retrieve the value of the "Beta" feature flag
    if(featureManager.isEnabledAsync(feature:"Beta").block()) {
            return "My feature is enabled!";
        } else {
            return "My feature is disabled.";
        }
```

GitHub Copilot

## Awareness

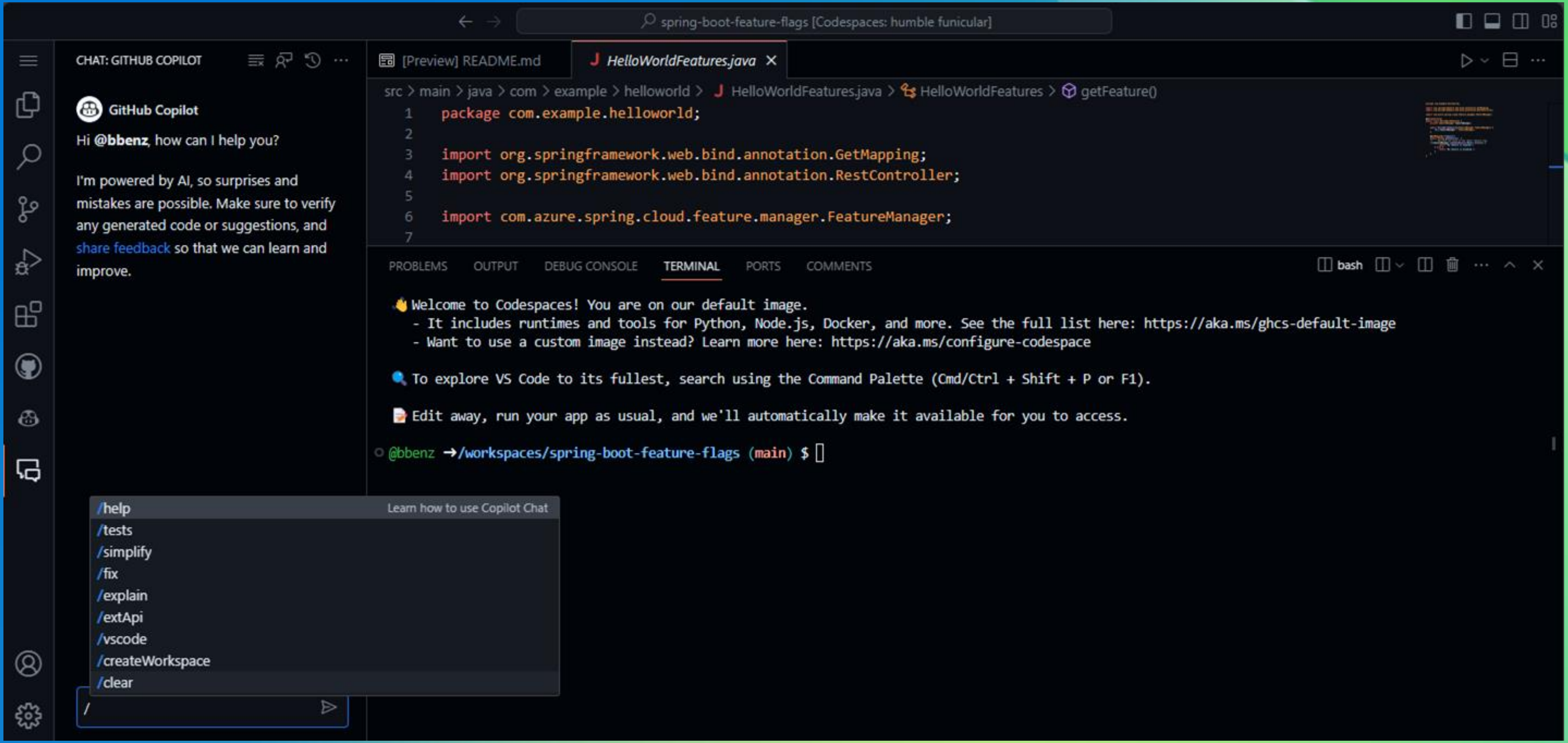
- File names
- Open files
- Cursor
- Selected



```java
package com.example.helloworld;

import org.springframework.boot.context.properties.ConfigurationProperties;

@ConfigurationProperties(prefix = "config")
public class MessageProperties {
    private String message;

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
}
```

# GitHub Copilot

## Codespaces support


spring-boot-feature-flags [Codespaces: humble funicular]
CHAT: GITHUB COPILOT
[Preview] README.md
HelloWorldFeatures.java
src > main > java > com > example > helloworld > HelloWorldFeatures.java > HelloWorldFeatures > getFeature()
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
bash
/help Learn how to use Copilot Chat
/tests
/simplify
/fix
/explain
/extApi
/vscode
/createWorkspace
/clear

GitHub Copilot

Hi **@bbenz**, how can I help you?

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve.

```java
package com.example.helloworld;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import com.azure.spring.cloud.feature.manager.FeatureManager;
```

```
👋 Welcome to Codespaces! You are on our default image.
   - It includes runtimes and tools for Python, Node.js, Docker, and more. See the full list here: https://aka.ms/ghcs-default-image
   - Want to use a custom image instead? Learn more here: https://aka.ms/configure-codespace

🔍 To explore VS Code to its fullest, search using the Command Palette (Cmd/Ctrl + Shift + P or F1).

📝 Edit away, run your app as usual, and we'll automatically make it available for you to access.

@bbenz →/workspaces/spring-boot-feature-flags (main) $ 
```

# DEMO

Automating OpenTelemetry integration and deployment scripts on AKS using GitHub Copilot

## Part 4: Practical Implementation and Visualization

# DEMO

Practical Implementation and Visualization

Microsoft

Thank you!
Questions?