vFunction

Support Matrix for Monolithic Apps

Java Support Matrix

OS	App Servers	JVMs / JDKs	Deployment Models
• Linux Alpine • Linux Debian • Linux CentOS 8+ • Linux Fedora • Linux RHEL 7+ • Linux Suse 12+ • Linux Ubuntu 14+ • Windows Server 2012 R2+	• Apache Tomcat 6+ • Eclipse Jetty 9.4.20+ • IBM WebSphere 8.5.5+ • IBM WebSphere Liberty 22+ • Open Liberty 21+ • Oracle WebLogic 10.3.6+ • Payara 5.2020+ • RedHat JBoss 7.2+ • RedHat Wildfly 14.0.1+	• Amazon Correto 1.8, 11, 17, 21 • Azul Zulu 8, 11, 17 • IBM J9 8.0.3+ • Oracle HotSpot / OpenJDK 1.6, 1.7, 1.8, 9, 11, 17, 21 • OpenJ9 1.8, 11, 17, 21 • SAPMachine 8, 11	• Docker, Kubernetes, OpenShift Containers running Linux images • Linux with Sudo or Sudoless access • Pivotal Cloud Foundry • Windows with PowerShell

.NET (C# and VB.NET) Support Matrix

OS	App	Startup Processes	Versions	Deployment Models
• Linux Amazon 2 • Linux CentOS 8+ • Linux openSUSE 15.1+ • Linux RHEL 8+ • Linux Ubuntu 18.04+ • Windows Server 2012 R2+	• Windows 32 bit • Windows 64 bit	• Linux Process • Linux Service • Windows Command Line • Windows IIS Web App • Windows Service	• Linux .NET 6.0+ • Windows .NET 6.0+ • Windows .NET Core 3.x • Windows .NET Framework 4.x	• Azure App Services • Docker, Kubernetes, OpenShift Containers running Linux or Windows Server images • Linux with Sudo or Sudoless access • Windows Server with PowerShell

Support Matrix for Distributed Apps

The following matrix summarizes the languages supported by OpenTelemetry (OTEL) and whether they can be used with or without code changes:

Language	**Supported Without Code Changes**	**Supported With Code Changes**
C++	No	Yes
.NET	Yes	Yes
Erlang/Elixir	No	Yes
Go	No	Yes
Java	Yes	Yes
JavaScript	Yes	Yes
PHP	Yes	Yes
Python	Yes	Yes
Ruby	No	Yes
Rust	No	Yes
Swift	No	Yes

As OTEL continues to evolve, support for additional languages and improvements to existing integrations are expected, all of which are expected to be supported by vFunction.

vFunction utilizes the OTEL capabilities in order to provide our users with a "Distributed Application" comprehensive view of the services that make up an application, their interactions, dependencies, and changes over time. The primary view for a Distributed Application is the Service Map Graph, a visual representation of the application architecture as a graph, where nodes represent services and edges represent dependencies and interactions between them.

To populate the Service Map Graph, vFunction leverages OTEL for trace collection and/or integrates with APM providers to fetch relevant service information. During defined learning periods, vFunction collects traces from the OTEL backend and queries the user's APM provider for service details. It processes the collected data into a single service map, providing a comprehensive view of the distributed application.