

BPA-Benutzerhandbuch Distributed Tracing Framework v5.1

- [Einleitung](#)
 - [Zipkin](#)
 - [@opentelemetry/api](#)
- [Wichtige Komponenten](#)
- [Voraussetzungen](#)
 - [Netzwerkanforderungen](#)
- [Docker ermöglicht Bereitstellung](#)
 - [Aktivieren des Überwachungsprofils](#)
 - [Überprüfen des Zipkin-Containers](#)
- [Aktivieren der Ablaufverfolgung in Services](#)
 - [Hierarchie der Konfigurationslast](#)
 - [Pod-spezifische Konfiguration](#)
 - [Anwendungsfall](#)
 - [Beispielszenario](#)
 - [Globale Konfiguration](#)
 - [Anwendungsfall](#)
 - [Beispielszenario](#)
 - [Fallback-Konfiguration](#)
 - [Anwendungsfall](#)
 - [Festlegen der Ablaufverfolgungskonfiguration](#)
 - [Konfigurationseigenschaften](#)
 - [Anwenden von Konfigurationsänderungen](#)
 - [Helm-Upgrade-Befehl](#)
- [Bereitstellung von Kubernetes und Helm](#)
 - [Aktivieren der Wertüberwachung](#)
 - [Bereitstellen des Zipkin-Diensts](#)
 - [Überprüfen der Bereitstellung](#)
 - [Überprüfen des Dienststatus](#)
- [Zugriff und Überwachung](#)
 - [1-Node-Zugriff auf Zipkin-Dashboard](#)
 - [3-Node Kubernetes Zugriff auf Zipkin Dashboard](#)
 - [Zipkin-Benutzeroberfläche](#)
- [Tracers und Spans](#)

Einleitung

Zipkin

Zipkin ist ein verteiltes Tracing-System, das hilft, Zeitdaten zu sammeln, die benötigt werden, um Latenzprobleme in der Service-Architektur zu beheben. Dieses Handbuch enthält Anweisungen zum Bereitstellen von Zipkin und zum Aktivieren der Ablaufverfolgung für BPA-Plattformdienste.

@opentelemetry/api

Das Paket `@opentelemetry/api` ist die Basis-API für OpenTelemetry in Node.js. Sie stellt Schnittstellen und Klassen zum Erstellen und Verwalten von Ablaufverfolgungen, Spannen und der Kontextanwendung bereit. Dieses Paket ermöglicht es Entwicklern, ihre Anwendungen zu instrumentieren, um Telemetriedaten wie Spuren und Metriken zu sammeln, die dann zur Analyse in Backends wie Zipkin exportiert werden können.

Wichtige Komponenten

Zu den Hauptkomponenten von Zipkin gehören:

- Zipkin-Server: Zentraler Ablaufverfolgungssammler und Benutzeroberfläche
- Offene Telemetrie: Instrumentationsbibliothek für die Ablaufverfolgung
- Elastische Suche: Speicher-Backend für Ablaufverfolgungsdaten
- Secure Sockets Layer (SSL) und Transport Layer Security (TLS): Sichere Kommunikation

Voraussetzungen

Netzwerkanforderungen

Um Zipkin bereitzustellen und die Ablaufverfolgung zu aktivieren, sind die folgenden Netzwerkanforderungen erforderlich:

- Port 9412: HTTPS-API-Endpunkt (Zipkin Hypertext Transfer Protocol Secure)
- Port 9411: Zipkin-HTTP-Management-Endpunkt (Hypertext Transfer Protocol)
- Port 9200: Zugriff auf das Cluster Elasticsearch

Docker ermöglicht Bereitstellung

Aktivieren des Überwachungsprofils

1. Starten Sie den BPA-Docker mit einem Knoten über den folgenden Pfad:

```
cd /opt/bpa/bpa-{build_version}/scripts
```

2. Führen Sie das Bash-Skript mit der Überwachungsoption aus. Verwenden Sie dazu den folgenden Befehl:

```
./startbpa.1node.sh monitoring
```

Überprüfen des Zipkin-Containers

1. Führen Sie den folgenden Befehl aus, um den Containerstatus zu überprüfen:

```
docker ps | grep tracers-zipkin
```

2. Um die Protokolle eines Zipkin-Containers anzuzeigen. Führen Sie den folgenden Befehl aus:

```
docker logs tracers-zipkin -f
```

Aktivieren der Ablaufverfolgung in Services

Die BPA-Anwendung unterstützt flexible Ablaufverfolgungskonfigurationen über Dateien mit der Bezeichnung "tracingConfig.json". Das System implementiert einen hierarchischen Mechanismus zum Laden von Konfigurationen mit drei (3) Prioritätsstufen für unterschiedliche Bereitstellungsszenarien.

Hierarchie der Konfigurationslast

Die Anwendung lädt die Ablaufverfolgungskonfigurationen basierend auf der Rangfolge in der folgenden Reihenfolge:

Pod-spezifische Konfiguration

Pfad: bpa-helm-chart/charts/<Dienstname>/public_conf

Anwendungsfall

- Individuelle, Pod-spezifische Ablaufverfolgungskonfiguration
- Ermöglicht unterschiedliche Ablaufverfolgungseinstellungen für bestimmte Dienste oder PODs
- Manuelle Platzierung erforderlich: Benutzer müssen diese Datei manuell im Pod-Ordnerpfad platzieren: bpa-helm-chart/charts/<Dienstname>/public_conf

Beispielszenario

- Detaillierte Ablaufverfolgung nur für kritische Services aktivieren
- Unterschiedliche Abtastraten für verschiedene Mikrodienste
- Debugspezifische Ablaufverfolgung zur Fehlerbehebung von PODs

Globale Konfiguration

Pfad: bpa-helm-chart/bpa/conf/common/globals/tracingConfig.json

Anwendungsfall

- Globale Ablaufverfolgungskonfiguration für alle Container und Pods
- Zentrales Tracing-Management
- Globale Auswirkungen: Änderungen an dieser Datei wirken sich auf alle Pods in der Bereitstellung aus

Beispielszenario

- Aktivieren und Deaktivieren der Ablaufverfolgung in der gesamten BPA-Bereitstellung
- Einheitliche Ablaufverfolgungsrichtlinien für die Produktionsumgebung festlegen
- Zentrale Compliance- und Überwachungsanforderungen

Fallback-Konfiguration

Pfad: ../conf/tracingConfig.json (Fallback)

Anwendungsfall

- Standardkonfiguration für die Ablaufverfolgung im Paket mit der Anwendung
- Standardeinstellungen für Entwicklungsumgebung
- Stellt sicher, dass die Anwendung nie aufgrund einer fehlenden Ablaufverfolgungskonfiguration fehlschlägt

Wenn eine Konfigurationsdatei falsch formatiert oder nicht lesbar ist, führt das System folgende Schritte aus:

- Protokolliert den Fehler
- Standardmäßig ist diese Option aktiviert: `false`}, um Ablaufverfolgungsfehler zu verhindern
- Fortsetzen des Anwendungsstarts

Festlegen der Ablaufverfolgungskonfiguration

Erstellen oder aktualisieren Sie die Datei "tracingConfig.json" mit dem folgenden Inhalt im Ordnerpfad des POD:

bpa-helm-chart/charts/<Dienstname>/public_conf/

Beispiele:

- So aktivieren Sie die Ablaufverfolgung

```
{  
  "enable": true  
}
```

- So deaktivieren Sie die Ablaufverfolgung:

```
{  
  "enable": false  
}
```

Konfigurationseigenschaften

- `enable` (boolean) (erforderlich): Steuert, ob die Ablaufverfolgung für die Anwendung aktiviert oder deaktiviert ist
- `Richtig`: Aktiviert verteilte Ablaufverfolgung
 - Die Anwendung initialisiert die Ablaufverfolgungshilfe.
 - Traces werden gesammelt und an das konfigurierte Backend gesendet
 - Leistungsüberwachung und aktivierte Debugging-Funktionen
- `Falsch`: Deaktiviert die verteilte Ablaufverfolgung
 - Kein Tracing-Overhead bei der Anwendungsleistung
 - Keine Erfassung oder Übertragung von Ablaufverfolgungsdaten
 - Anwendung wird ohne Ablaufverfolgungsinstrumentation ausgeführt

Anwenden von Konfigurationsänderungen

Helm-Upgrade-Befehl

Nachdem Sie die Dateien "tracingConfig.json" an den entsprechenden Speicherorten platziert haben, wenden Sie die Änderungen mithilfe des Befehls `helm upgrade` (Helm-Upgrade) an:

```
helm upgrade bpa-rel --namespace bpa-ns /opt/bpa-helm-chart
```

Befehlsaufschlüsselung:

- `helm upgrade`: Aktualisiert eine vorhandene Helm-Version
- `bpa-rel`: Name der BPA Helm-Version
- `—namespace bpa-ns`: Gibt den Kubernetes-Namespace an, in dem BPA bereitgestellt wird.
- `/opt/bpa-helm-chart`: Pfad zum BPA Helm-Diagrammverzeichnis

Bereitstellung von Kubernetes und Helm

Aktivieren der Wertüberwachung

Bearbeiten Sie im folgenden Helm-Diagrammpfad die Helm-Werte in der Datei "values.yaml", um die Überwachung zu aktivieren:

```
yaml
  global:
    enableMonitoring: true
```

Bereitstellen des Zipkin-Diensts

Stellen Sie den Zipkin-Dienst mithilfe des folgenden Helm-Befehls bereit:

```
helm install bpa-rel --create-namespace --namespace bpa-ns
```

Überprüfen der Bereitstellung

Verwenden Sie die folgenden Befehle, um die Bereitstellung zu überprüfen:

```
kubectl get pods -n bpa-ns | grep tracers-zipkin
kubectl get svc -n bpa-ns | grep tracers-zipkin
```

Überprüfen des Dienststatus

Verwenden Sie den folgenden Befehl, um POD-Protokolle anzuzeigen:

```
kubectl logs -n bpa-ns deployment/tracers-zipkin -f
```

Verwenden Sie den folgenden Befehl, um Dienstendpunkte zu überprüfen:

```
kubectl describe svc tracers-zipkin -n bpa-ns
```

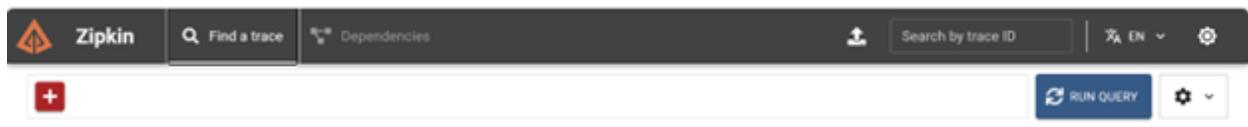
Zugriff und Überwachung

1-Node-Zugriff auf Zipkin-Dashboard

URL: <https://<SERVER IP>:9412/zipkin/>

3-Node Kubernetes Zugriff auf Zipkin Dashboard

URL: <https://<cluster-ip>:30900/zipkin/>



Search Traces

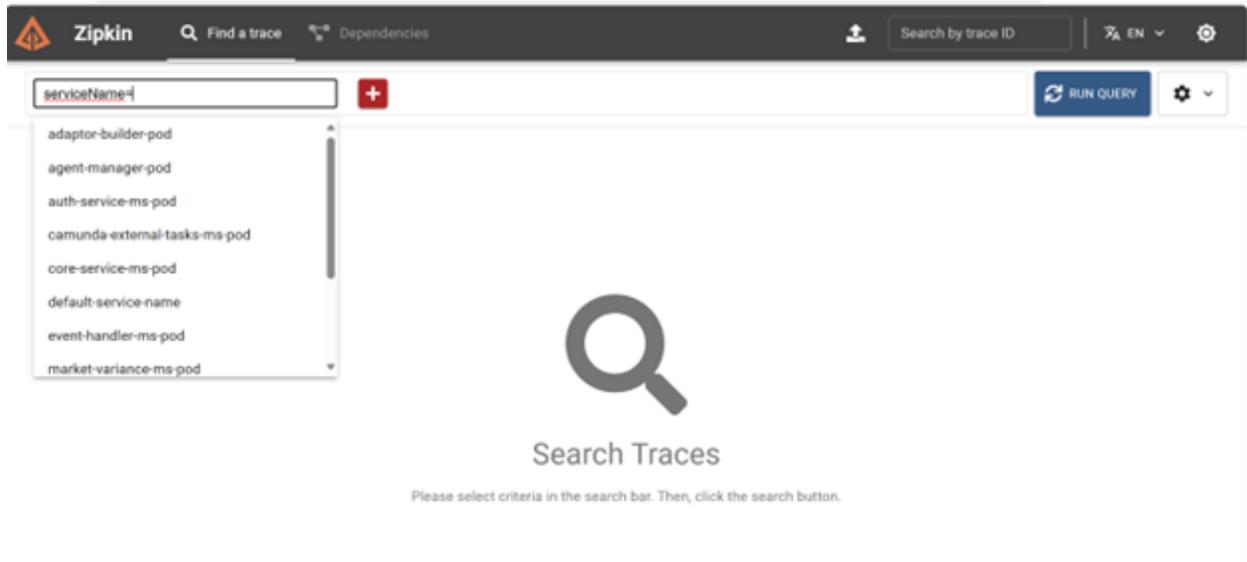
Please select criteria in the search bar. Then, click the search button.

Zipkin-Dashboard

Zipkin-Benutzeroberfläche

Das Zipkin-Dashboard bietet eine Benutzeroberfläche zum Suchen und Visualisieren von Spuren. Zu den Hauptkomponenten gehören:

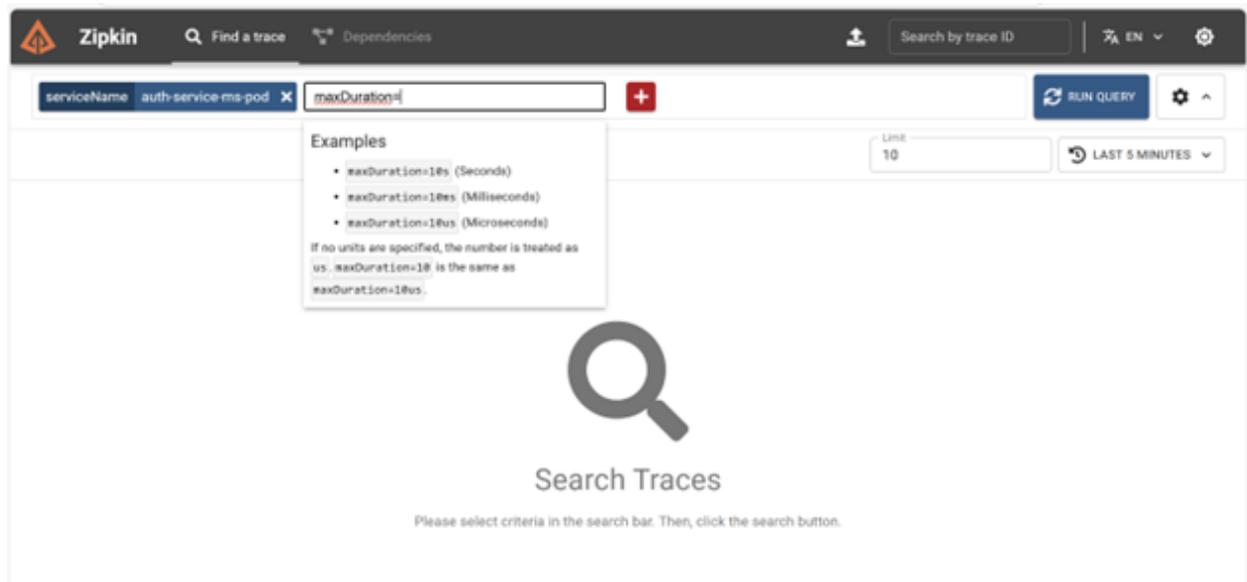
- Service-Name-Dropdown: Nachverfolgungen nach Dienstnamen filtern



ServiceName

ServiceName

- Zeitbereichsauswahl: Zeitfenster für die Trace-Suche definieren



Zeitbereichsauswahl

- Nachverfolgungsliste: Zeigt eine Liste der Ablaufverfolgungen an, die den Suchkriterien entsprechen.

The screenshot shows the Zipkin dashboard with the service name 'auth-service-ms-pod' selected. The results table lists various spans with their start times, spans, and durations. The durations are visualized as horizontal bars.

Root	Start Time	Spans	Duration
auth-service-ms-pod: pg connect	a minute ago (07/18 08:38:33:184)	1	13.373ms
auth-service-ms-pod: pg connect	2 minutes ago (07/18 08:37:49:631)	1	13.353ms
auth-service-ms-pod: get /api/v[0-9]+([0-9]+)/internal/tenantsinheaders	2 minutes ago (07/18 08:37:45:107)	26	5.936ms
auth-service-ms-pod: get /	a minute ago (07/18 08:38:25:240)	23	2.556ms
auth-service-ms-pod: pg.query/select kong	2 minutes ago (07/18 08:37:33:198)	1	740.000µs
auth-service-ms-pod: pg.query/set kong	2 minutes ago (07/18 08:37:49:644)	1	704.000µs
auth-service-ms-pod: pg.query/set kong	2 minutes ago (07/18 08:37:33:198)	1	570.000µs
auth-service-ms-pod: pg.query/set kong	a minute ago (07/18 08:38:33:197)	1	576.000µs
auth-service-ms-pod: pg.query/select kong	a minute ago (07/18 08:38:33:198)	1	468.000µs
auth-service-ms-pod: pg.query/select kong	2 minutes ago (07/18 08:37:49:645)	1	443.000µs

Ablaufverfolgungsliste

- Ablaufverfolgungszeitleiste: Visuelle Darstellung von Zeitintervallen (d. h. Spannen) innerhalb einer Spur

The screenshot shows the Zipkin dashboard with the search trace modal open. The modal displays a list of lookback intervals and two lookback configuration sections: 'Millis Lookback' and 'Range Lookback'.

Lookback Intervals:

- Last 1 minute
- Last 5 minutes
- Last 15 minutes
- Last 30 minutes
- Last 1 hour
- Last 2 hours
- Last 3 hours
- Last 6 hours
- Last 12 hours

Millis Lookback:

Milliseconds: **APPLY**

Range Lookback:

Start Time: **APPLY**

End Time: **APPLY**

Ablaufverfolgungszeitleiste

- Spannweitendetails: Detaillierte Ansicht der einzelnen Bereiche, einschließlich Dauer, Tags und Protokolle.

Span ID	Service name	Span name	Start time	Duration
48df1e037faa26	auth-service-ms-pod	get /api/v[0-9]+/[0-9]+[07/18 08:37:45.107	5.936ms
0a2b964630c0f2	auth-service-ms-pod	middleware - anonymo..	07/18 08:37:45.107	12.000µs
4590fb36337736	auth-service-ms-pod	middleware - anonymo..	07/18 08:37:45.107	467.000µs
633ab670a89b1b0	auth-service-ms-pod	middleware - anonymo..	07/18 08:37:45.107	8.000µs
2a28c6577647910f	auth-service-ms-pod	middleware - cookiepars	07/18 08:37:45.107	26.000µs
c83796abd2f1b956	auth-service-ms-pod	middleware - corsmiddle..	07/18 08:37:45.107	70.000µs
0c79a309b1e749b4	auth-service-ms-pod	middleware - expressint	07/18 08:37:45.107	57.000µs
da6798c5c8aa0444	auth-service-ms-pod	middleware - jsonpars	07/18 08:37:45.107	15.000µs
50a2619ac283995	auth-service-ms-pod	middleware - middleware	07/18 08:37:45.107	15.000µs
c75eaa82679b57c	auth-service-ms-pod	middleware - query	07/18 08:37:45.107	43.000µs
b697d9582392a6c	auth-service-ms-pod	middleware - urlencoded..	07/18 08:37:45.107	40.000µs
0893aa480a99ef	auth-service-ms-pod	middleware - anonymo..	07/18 08:37:45.108	31.000µs
4a04cb7104b66807	auth-service-ms-pod	middleware - anonymo..	07/18 08:37:45.108	121.000µs
08ba7306d43013	auth-service-ms-pod	middleware - anonymo..	07/18 08:37:45.108	50.000µs
9690e40cc6036e0	auth-service-ms-pod	middleware - anonymo..	07/18 08:37:45.108	19.000µs
b05ed4820ca6288	auth-service-ms-pod	middleware - anonymo..	07/18 08:37:45.108	24.000µs
c638a16b2833cb0	auth-service-ms-pod	middleware - anonymo..	07/18 08:37:45.108	53.000µs
7e19ea50040e2179	auth-service-ms-pod	middleware - initializ	07/18 08:37:45.108	74.000µs
a20a9a90a2650c47	auth-service-ms-pod	router - /about/cisco-bp..	07/18 08:37:45.108	3.000µs
c0e72796c495b7a	auth-service-ms-pod	router - /about/cisco-bp..	07/18 08:37:45.108	4.000µs

Details zur Spanne

Tracers und Spans

Tracer sind für das Erstellen und Verwalten von Spans zuständig. Ein Tracer ist einem bestimmten Dienst oder einer Anwendungskomponente zugeordnet. Spannweiten stellen eine einzelne Arbeits- oder Betriebseinheit innerhalb einer Spur dar. Jede Spanne enthält Informationen wie den Namen des Vorgangs, die Start- und Endzeit, Attribute und Beziehungen zwischen über- und untergeordneten Elementen und anderen Spans.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.