

# Guía del usuario de BPA sobre Distributed Tracing Framework v5.1

- [Introducción](#)
  - [Cremallera](#)
  - [@opentelemetry/api](#)
- [Componentes clave](#)
- [Prerequisites](#)
  - [Requisitos de la red](#)
- [Docker crea la implementación](#)
  - [Activación del perfil de supervisión](#)
  - [Verificación del contenedor Zipkin](#)
- [Habilitación del seguimiento en los servicios](#)
  - [Jerarquía de carga de configuración](#)
    - [Configuración específica de POD](#)
      - [caso de uso](#)
      - [Situación de ejemplo](#)
    - [Configuración global](#)
      - [caso de uso](#)
      - [Situación de ejemplo](#)
    - [Configuración de reserva](#)
      - [caso de uso](#)
  - [Establecer configuración de seguimiento](#)
    - [Propiedades de configuración](#)
  - [Aplicación de cambios de configuración](#)
    - [Comando Helm Upgrade](#)
- [Implementación de Kubernetes y Helm](#)
  - [Habilitación de la supervisión en valores](#)
  - [Implementación del servicio Zipkin](#)
  - [Verificación de la implementación](#)
  - [Comprobación del estado del servicio](#)
- [Acceso y supervisión](#)
  - [Acceso de 1 nodo al panel de cremallera](#)
  - [Acceso de Kubernetes de 3 nodos al panel de Zipkin](#)
  - [Uso de Zipkin UI](#)
- [Trazadores y extensiones](#)

## Introducción

### Cremallera

Zipkin es un sistema de seguimiento distribuido que ayuda a recopilar los datos de tiempo necesarios para solucionar los problemas de latencia en la arquitectura de servicio. Esta guía proporciona instrucciones para implementar Zipkin y habilitar el seguimiento en los servicios de la plataforma BPA.

## @opentelemetry/api

El paquete `@opentelemetry/api` es la API principal para OpenTelemetry en Node.js. Proporciona interfaces y clases para crear y administrar seguimientos, extensiones y propagación de contexto. Este paquete permite a los desarrolladores instrumentar sus aplicaciones para recopilar datos de telemetría, como rastros y métricas, que luego se pueden exportar a motores como Zipkin para su análisis.

## Componentes clave

Los componentes clave de Zipkin incluyen:

- Servidor Zipkin: Recopilador de seguimiento central e interfaz de usuario (IU)
- Telemetría abierta: Biblioteca de instrumentación para seguimiento
- Elasticsearch: Motor de almacenamiento para datos de seguimiento
- Secure Sockets Layer (SSL) y Transport Layer Security (TLS): Proporciona una comunicación segura

## Prerequisites

### Requisitos de la red

Para implementar Zipkin y habilitar el seguimiento, se requieren los siguientes requisitos de red:

- Port 9412: Extremo API de protocolo de transferencia de hipertexto seguro (HTTPS) Zipkin
- Port 9411: Extremo de administración del Protocolo de transferencia de hipertexto (HTTP) Zipkin
- Port 9200: Acceso al clúster Elasticsearch

## Docker crea la implementación

### Activación del perfil de supervisión

1. Inicie el acoplador de 1 nodo BPA desde la siguiente ruta:

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

2. Ejecute la secuencia de comandos bash con la opción de supervisión mediante el siguiente comando:

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

## Verificación del contenedor Zipkin

1. Para comprobar el estado del contenedor, ejecute el siguiente comando:

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

2. Para ver los registros de un contenedor Zipkin. ejecute el siguiente comando:

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

## Habilitación del seguimiento en los servicios

La aplicación BPA admite una configuración de seguimiento flexible a través de los archivos "traceConfig.json". El sistema implementa un mecanismo de carga de configuración jerárquico con tres (3) niveles de precedencia para dar cabida a diferentes escenarios de implementación.

### Jerarquía de carga de configuración

La aplicación carga las configuraciones de seguimiento en el orden siguiente en función de la precedencia:

Configuración específica de POD

Ruta: bpa-helm-chart/charts/<service name>/public\_conf

caso de uso

- Configuración de seguimiento individual específica de grupo de dispositivos
- Permite una configuración de seguimiento diferente para servicios o grupos de dispositivos específicos
- Colocación manual requerida: Los usuarios deben colocar manualmente este archivo en la ruta de la carpeta del grupo de dispositivos: bpa-helm-chart/charts/<service name>/public\_conf

Situación de ejemplo

- Habilitar el seguimiento detallado solo para servicios críticos
- Diferentes tasas de muestreo para diferentes microservicios
- Seguimiento específico de depuración para grupos de solución de problemas

Configuración global

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

caso de uso

- Configuración de seguimiento global para todos los contenedores y grupos de dispositivos
- Administración de seguimiento centralizada
- Impacto global: Los cambios realizados en este archivo afectan a todos los grupos de dispositivos de la implementación

Situación de ejemplo

- Habilitar y deshabilitar el seguimiento en toda la implementación de BPA
- Establecer políticas de seguimiento uniformes para el entorno de producción
- Cumplimiento y requisitos de supervisión centralizados

Configuración de reserva

Ruta: ../conf/tracingConfig.json (reserva)

## caso de uso

- Configuración de seguimiento predeterminada incluida con la aplicación
- Valores predeterminados del entorno de desarrollo
- Garantiza que la aplicación nunca falle debido a una configuración de seguimiento que falta

Si algún archivo de configuración está mal formado o es ilegible, el sistema:

- Registra el error
- El valor predeterminado es {enable: false} para evitar errores de seguimiento
- Continúa el inicio de la aplicación

## Establecer configuración de seguimiento

Cree o actualice el archivo "traceConfig.json" con el siguiente contenido en la ruta de la carpeta del grupo de dispositivos:

bpa-helm-chart/charts/<service name>/public\_conf/

Examples:

- Para habilitar el seguimiento

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

- Para deshabilitar el seguimiento:

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

## Propiedades de configuración

- `enable (boolean)(required)`: Controla si el seguimiento está habilitado o deshabilitado para la aplicación
- `verdadero`: Habilita el seguimiento distribuido
  - La aplicación inicializa el ayudante de seguimiento
  - Los seguimientos se recopilan y se envían al backend configurado
  - Capacidades de depuración y supervisión del rendimiento activadas
- `falso`: Deshabilita el seguimiento distribuido
  - Sin sobrecarga de seguimiento en el rendimiento de la aplicación
  - Sin recopilación ni transmisión de datos de seguimiento
  - La aplicación se ejecuta sin instrumentación de seguimiento

## Aplicación de cambios de configuración

### Comando Helm Upgrade

Después de colocar los archivos "traceConfig.json" en las ubicaciones apropiadas, aplique los cambios mediante el comando Helm upgrade:

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

### Desglose de comandos:

- `actualización de helm`: Actualiza una versión de Helm existente
- `bpa-rel`: Nombre de la versión de BPA Helm
- `—namespace bpa-ns`: Especifica el espacio de nombres de Kubernetes donde se implementa BPA
- `/opt/bpa-helm-chart`: Ruta de acceso al directorio de gráficos de BPA Helm

## Implementación de Kubernetes y Helm

### Habilitación de la supervisión en valores

En la siguiente ruta del gráfico del timón, edite los valores del timón en el archivo "values.yaml" para habilitar la supervisión:

yaml

```
global:
enableMonitoring: true
```

## Implementación del servicio Zipkin

Implemente el servicio Zipkin con el siguiente comando Helm:

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

## Verificación de la implementación

Utilice los siguientes comandos para verificar la implementación:

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

## Comprobación del estado del servicio

Para ver los registros de grupo de dispositivos, utilice el siguiente comando:

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

Para comprobar los extremos del servicio, utilice el siguiente comando:

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

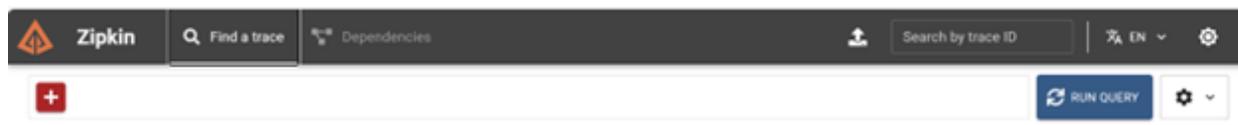
## Acceso y supervisión

Acceso de 1 nodo al panel de cremallera

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

Acceso de Kubernetes de 3 nodos al panel de Zipkin

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



Search Traces

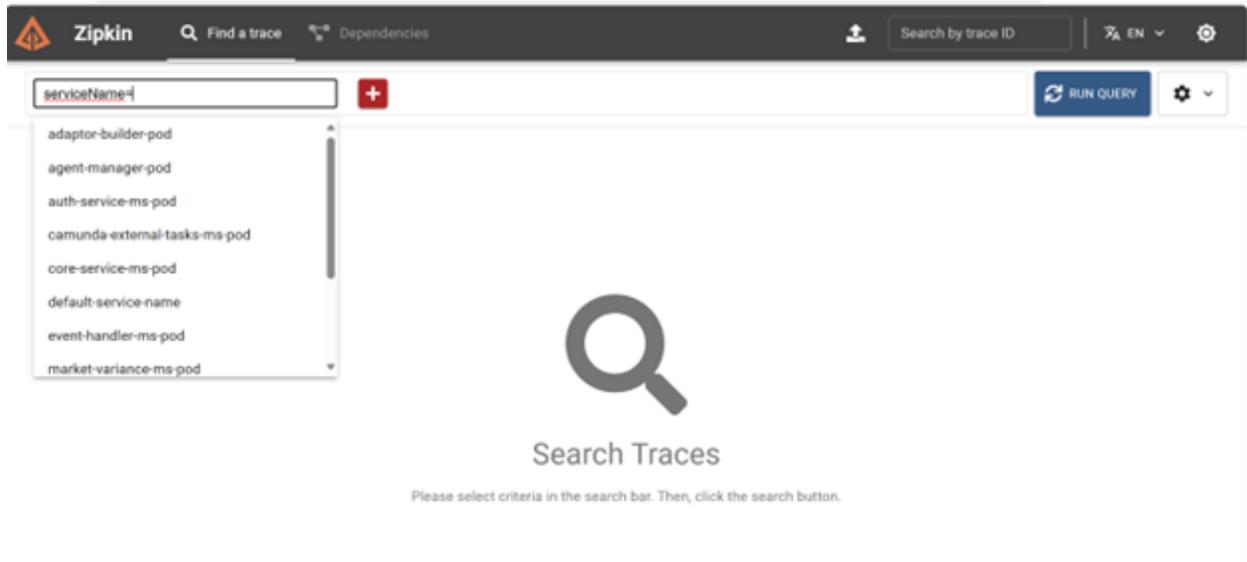
Please select criteria in the search bar. Then, click the search button.

Panel de cierre

## Uso de Zipkin UI

El panel de Zipkin proporciona una interfaz de usuario para buscar y visualizar rastros. Los componentes clave son:

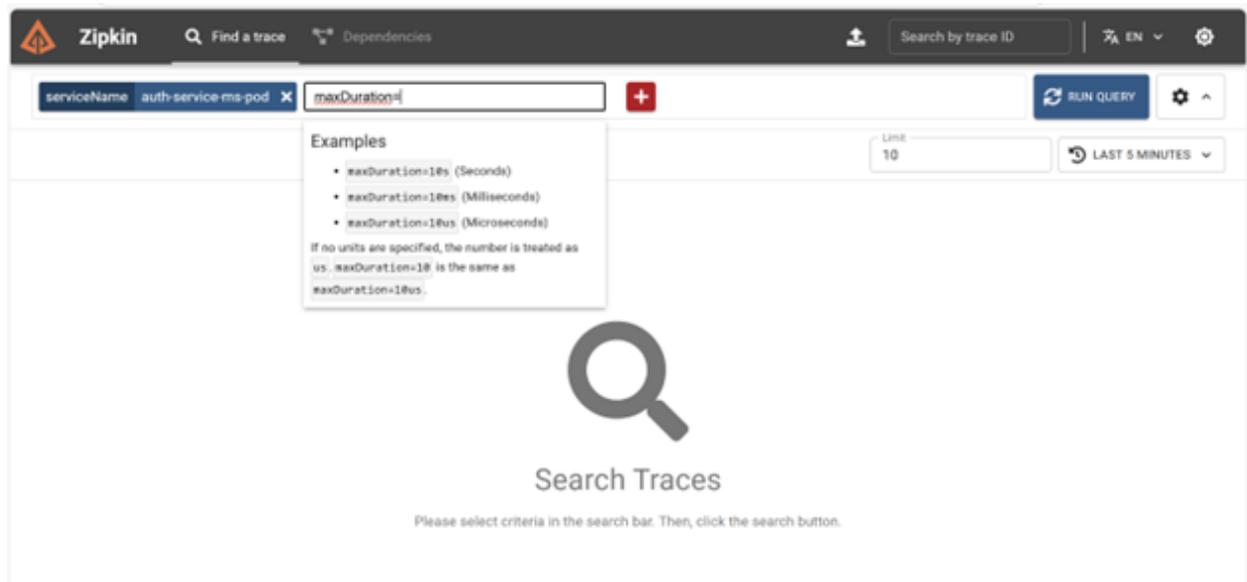
- Menú desplegable Nombre de servicio: Filtrar seguimientos por nombre de servicio



Nombre del servicio

Nombre del servicio

- Selector de rango de tiempo: Definir la ventana de tiempo para la búsqueda de seguimiento



Selector de rango de tiempo

- Lista de seguimiento: Muestra una lista de los seguimientos que coinciden con los criterios de búsqueda

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.

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

### Lista de seguimiento

- Línea de tiempo de seguimiento: Representación visual de intervalos de tiempo (es decir, span) dentro de un seguimiento

The screenshot shows the Zipkin dashboard with the search trace modal open. The modal includes a search bar, a time range selector, and a range lookback section.

**Search Trace**

Please select criteria in the search bar. Then, click on the search icon.

**Time Range Selector:**

- 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

**Range Lookback:**

Start Time: 07/18/2025 07:38:16

End Time: 07/18/2025 08:38:16

### Línea de tiempo de seguimiento

- Detalles del tramo: Vista detallada de cada tramo, incluida la duración, las etiquetas y los registros.

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
533ab670a89b1b0	auth-service-ms-pod	middleware - anonymo...	07/18 08:37:45.107	8.000µs
2a28c0577647910f	auth-service-ms-pod	middleware - cookieparser	07/18 08:37:45.107	26.000µs
c83796bd2f1b956	auth-service-ms-pod	middleware - corsmiddle...	07/18 08:37:45.107	70.000µs
0c7a309b1a749b4	auth-service-ms-pod	middleware - expressint	07/18 08:37:45.107	57.000µs
da6798c5c8aa0444	auth-service-ms-pod	middleware - jsonparser	07/18 08:37:45.107	15.000µs
50a2619ac283995	auth-service-ms-pod	middleware - middleware	07/18 08:37:45.107	15.000µs
c75eab2679b57c	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
08a07306d43013	auth-service-ms-pod	middleware - anonymo...	07/18 08:37:45.108	50.000µs
9690a40cc6030e0	auth-service-ms-pod	middleware - anonymo...	07/18 08:37:45.108	19.000µs
b05ed4820ca288	auth-service-ms-pod	middleware - anonymo...	07/18 08:37:45.108	24.000µs
c538a16b28330b0	auth-service-ms-pod	middleware - anonymo...	07/18 08:37:45.108	53.000µs
7a19ea5040a2179	auth-service-ms-pod	middleware - initialize	07/18 08:37:45.108	74.000µs
a20a9a90a2650c47	auth-service-ms-pod	router - /about/cisco-tp...	07/18 08:37:45.108	3.000µs
c0e72796c495b7a	auth-service-ms-pod	router - /about/cisco-tp...	07/18 08:37:45.108	4.000µs

Detalles del intervalo

## Trazadores y extensiones

Los rastreadores son responsables de crear y administrar los tramos. Un rastreador está asociado a un servicio o componente de aplicación específico. Los tramos representan una sola unidad de trabajo u operación dentro de un seguimiento. Cada tramo contiene información como el nombre de la operación, la hora de inicio y finalización, los atributos y las relaciones principal-secundario con otros tramos.

## Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).