Caution

The SignalFx Tracing Library for PHP is deprecated as of February 21, 2024 and will reach End of Support (EOS) on February 21 2025. Until then, only critical security fixes and bug fixes will be provided. After the EOS date, the library will be archived and no longer maintained.

If you want to instrument new or existing PHP applications, use OpenTelemetry PHP instrumentation, which offers similar functionalities.

Instrument your PHP application for Splunk Observability Cloud (deprecated)

The SignalFx Tracing Library for PHP automatically instruments PHP applications.

To get started, use the guided setup or follow the instructions manually.

Generate customized instructions using the guided setup

To generate all the basic installation commands for your environment and application, use the PHP guided setup. To access the PHP guided setup, follow these steps:

1. Log in to Splunk Observability Cloud.
2. Open the PHP guided setup . Optionally, you can navigate to the guided setup on your own:
    1. In the navigation menu, select **Data Management**.
    2. Go to the **Available integrations** tab, or select **Add Integration** in the **Deployed integrations** tab.
    3. In the integration filter menu, select **By Product**.
    4. Select the **APM** product.
    5. Select the **PHP** tile to open the PHP guided setup.

Install the SignalFx Tracing Library for PHP manually

If you don’t use the guided setup, follow these instructions to manually install the SignalFx Tracing Library for PHP:

1. Instrument your PHP application
2. Deploy the PHP instrumentation in your environment
3. Configure the PHP instrumentation for Splunk Observability Cloud

Instrument your PHP application

Follow these steps to automatically instrument your application:

1. Check that you meet the requirements. See PHP instrumentation compatibility and requirements (deprecated).
2. Download the installation script from the following location:
   ```
   curl -LO  https://github.com/signalfx/signalfx-php-tracing/releases/latest/download/signalfx-setup.php
   ```
3. Install by running the installation script:
   ```
   php signalfx-setup.php --php-bin=all
   ```

   ### Note

   If you omit the `--php-bin` option, you can interactively select the PHP installation.

4. Set the following environment variables:

   ### Apache configuration

   ```
   # Add the following lines to your Apache configuration file

   SetEnv SIGNALFX_SERVICE_NAME="<my-service-name>"
   SetEnv SIGNALFX_ENDPOINT_URL='http://localhost:9411/api/v2/spans'
   SetEnv SIGNALFX_TRACE_GLOBAL_TAGS="deployment.environment:<my_environment>"
   ```

   ### Terminal

   ```
   export SIGNALFX_SERVICE_NAME="<my-service-name>"
   export SIGNALFX_ENDPOINT_URL='http://localhost:9080/v1/trace'
   export SIGNALFX_TRACE_GLOBAL_TAGS="deployment.environment:<my_environment>"
   ```

   Set environment variables globally or using the start script of your PHP application.

5. Restart your server.

Next, deploy the PHP instrumentation in your environment. See Deploy the PHP instrumentation in your environment for more information.

Note

If you need to add custom attributes to spans or want to manually generate spans, instrument your PHP application or service manually. See Manually instrument PHP applications for Splunk Observability Cloud.

INI file settings

If you don’t set any environment variable, the library extracts default values from the INI file. The prefix for settings defined using environment variables that start with `SIGNALFX_TRACE_` is `signalfx.trace.`. For all other environment variables that start with `SIGNALFX_` the prefix is `signalfx.`.

You can use the `signalfx-setup.php` script to set INI file options without having to manually locate each file. For example:

```
php signalfx-setup.php --update-config --signalfx.endpoint_url=http://172.17.0.1:9080/v1/trace
```

This is useful for options common to all PHP services running in the system, like endpoints.

Deploy the PHP instrumentation in your environment

You can deploy the PHP instrumentation in Docker or, Kubernetes, or you can send data directly to Splunk Observability Cloud. See the following sections for instructions for your environment:

- Deploy the PHP instrumentation in Docker
- Deploy the PHP instrumentation in Kubernetes
- Send data directly to Splunk Observability Cloud

Deploy the PHP instrumentation in Docker

You can deploy the PHP instrumentation using Docker. Follow these steps to get started:

1. Create a startup shell script in a location Docker can access. The script can have any name, for example setup.sh.
2. Edit the startup shell script to export the environment variables described in Instrument your PHP application.
3. Add the following commands to the startup shell script to initialize the PHP instrumentation:

   ```
   curl -LO https://github.com/signalfx/signalfx-php-tracing/releases/latest/download/signalfx-setup.php


   php signalfx-setup.php --php-bin=all
   php signalfx-setup.php --update-config
   --signalfx.endpoint_url=https://ingest.<realm>.signalfx.com/v2/trace/signalfxv1
   php signalfx-setup.php --update-config --signalfx.access_token=<access_token>
   php signalfx-setup.php --update-config --signalfx.service_name=<service-name>
   ```

4. Add a line to the script to start the application using the `supervisorctl`, `supervisord`, `systemd`, or a similar command. The following example uses `supervisorctl`:

   ```
   supervisor start my-php-app
   ```

5. Add a command to run the newly created shell script at the end of the Dockerfile.
6. Rebuild the container using the `docker build` command.

Next, configure the PHP instrumentation for Splunk Observability Cloud. See Configure the PHP instrumentation for Splunk Observability Cloud.

Caution

Make sure to deactivate the `Xdebug` extension, as it’s not compatible with the PHP instrumentation.

Deploy the PHP instrumentation in Kubernetes

To deploy the PHP instrumentation in Kubernetes, configure the Kubernetes Downward API to expose environment variables to Kubernetes resources.

The following example shows how to update a deployment to expose environment variables by adding the agent configuration under the `.spec.template.spec.containers.env` section:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
```

```yaml
  name: my-deployment
spec:
  selector:
    matchLabels:
      app: your-application
  template:
    metadata:
      labels:
        app: your-application
    spec:
      containers:
      - name: myapp
        image: <image-name>
        env:
          - name: HOST_IP
            valueFrom:
              fieldRef:
                fieldPath: status.hostIP
          - name: SIGNALFX_SERVICE_NAME
            value: "<service-name>"
          - name: SIGNALFX_ENDPOINT_URL
            value: "http://$(HOST_IP):9411/api/v2/spans"
          - name: SIGNALFX_TRACE_GLOBAL_TAGS
            value: "deployment.environment:<my_environment>"
```

Next, configure the PHP instrumentation for Splunk Observability Cloud. See Configure the PHP instrumentation for Splunk Observability Cloud.

Send data directly to Splunk Observability Cloud

By default, all telemetry is sent to the local instance of the Splunk Distribution of OpenTelemetry Collector.

To bypass the OTel Collector and send data directly to Splunk Observability Cloud, set the following environment variables:

Apache configuration

```
SetEnv SIGNALFX_ACCESS_TOKEN=<access_token>
SetEnv SIGNALFX_ENDPOINT_URL=https://ingest.<realm>.signalfx.com/v2/trace/signalfxv1
```

Terminal

```
export SIGNALFX_ACCESS_TOKEN=<access_token>
export SIGNALFX_ENDPOINT_URL=https://ingest.<realm>.signalfx.com/v2/trace/signalfxv1
```

To override the host used by the agent, use the environment variable `OTEL_RESOURCE_ATTRIBUTES` to set your host’s name to the desired source:

Windows PowerShell

```
$env:OTEL_RESOURCE_ATTRIBUTES=host.name=<host_name>
```

Linux

```
export OTEL_RESOURCE_ATTRIBUTES=host.name=<host_name>
```

To obtain an access token, see Retrieve and manage user API access tokens using Splunk Observability Cloud.

To find your Splunk realm, see Note about realms.

Next, configure the PHP instrumentation for Splunk Observability Cloud. See Configure the PHP instrumentation for Splunk Observability Cloud for more information.

Note

For more information on the ingest API endpoints, see Send APM traces .