

## Dapr Observability

### Distributed Tracing and Log Export Challenges

**Issue 1: Distributed Tracing** We are experiencing challenges exporting traces to **Zipkin** (for local development) and **Application Insights** (for the production environment) for operations performed by the Dapr runtime, including:

1. Get Secrets
2. Get/Set State
3. Service Invocation
4. Pub/Sub

**Expected Behavior:** All traces should be exported to Zipkin in local environments and ApplInsights in production. For example, traces should be generated and viewable for service invocation.

#### Setup Details:

- **Dapr Version:**
  - Runtime: 1.15.3
  - CLI: 1.15.0-rc.5
- **Architecture:** Dapr-based microservices using ASP.NET Core Web API. Applications are deployed/running via Docker Compose.

**Zipkin Setup:** We see traces for operations like *Get Secrets* and *Get/Set State*, but not for *Service Invocation*. Below is the configuration

Note: The yaml files referenced here are available in this public repository. - <https://github.com/nsathappan/dapr-observability> )

Config.yaml <https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/zipkin/config.yaml>

```
! config.yaml X
C: > Source > Repos > dapr-observability > yaml files > zipkin > ! config.yaml
1  apiVersion: dapr.io/v1alpha1
2  kind: Configuration
3  metadata:
4    name: daprConfig
5    namespace: default
6  spec:
7    mtls:
8      enabled: false
9    tracing:
10     samplingRate: "1"
11     zipkin:
12       endpointAddress: "http://host.docker.internal:9411/api/v2/spans"
```

Note: With docker compose setup, accessing localhost:9411 does not work; however, using host.docker.internal:9411 allows us to see some traces.

Localhost works only when we use dapr selfhosted mode.

docker-compose.yaml <https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/zipkin/docker-compose.yaml>

```
docker-compose.yaml X
C: > Source > Repos > dapr-observability > yaml files > zipkin > docker-compose.yaml
1  services:
2    app1Service:
3      image: ${DOCKER_REGISTRY-}app1
4      build:
5        context: .
6        dockerfile: App1/Dockerfile
7      ports:
8        - "40000:41000"
9
10   app1Service-sidecar:
11     image: "daprio/daprd:latest"
12     command: [
13       "./daprd",
14       "-app-id",
15       "app1-Service",
16       "-app-port", "41000",
17       "-components-path", "/components",
18       "--config", "/components/config.yaml",
19       "-log-level", "debug"
20     ]
21     volumes:
22       - "./components:/components"
23     depends_on:
24       - app1Service
25     network_mode: "service:app1Service"
26
27   app2Service:
28     image: ${DOCKER_REGISTRY-}app2
29     build:
30       context: .
31       dockerfile: App2/Dockerfile
32     ports:
33       - "40001:41001"
34
35   app2Service-sidecar:
36     image: "daprio/daprd:latest"
37     command: [
38       "./daprd",
39       "-app-id",
40       "app2-Service",
41       "-app-port", "41001",
42       "-components-path", "/components",
43       "--config", "/components/config.yaml",
44       "-log-level", "debug"
45     ]
46     volumes:
47       - "./components:/components"
48     depends_on:
```

**Application Insights Setup:** We attempted to export traces to AppInsights using the following configurations, but neither approach worked:

### Option 1: AppInsights Exporter

config\_appinsights\_1.yaml ( [https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/exporters/config\\_appinsights\\_1.yaml](https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/exporters/config_appinsights_1.yaml) )

```
! config_appinsights.yaml X
C: > Source > Repos > dapr-observability > yaml files > exporters > ! config_appinsights.yaml
1  apiVersion: dapr.io/v1alpha1
2  kind: Configuration
3  metadata:
4    name: daprConfig
5  spec:
6    tracing:
7      samplingRate: "1.0"
8    zipkin:
9      endpointAddress: "https://dc.services.visualstudio.com/v2/track"
```

config\_appInsights\_2.yaml ( [https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/exporters/config\\_appinsights\\_1.yaml](https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/exporters/config_appinsights_1.yaml) )

```
! config_appInsights_2.yaml ●
C: > Source > Repos > dapr-observability > yaml files > exporters > ! config_appInsights_2.yaml
1  apiVersion: dapr.io/v1alpha1
2  kind: Configuration
3  metadata:
4    name: appconfig
5  spec:
6    tracing:
7      enabled: true
8      exporter:
9        type: azure-appinsights
10       config:
11         instrumentationKey: <<YOUR INSTRUMENTATION KEY>>
12         endpoint: https://eastus-8.in.applicationinsights.azure.com/
13 components:
14   exporters:
15     azure-appinsights:
16       type: exporters.azure.appinsights
17       version: v1
```

## Option 2: OpenTelemetry collector

**Config.yaml:** ( [https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/otel/config\\_otel.yaml](https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/otel/config_otel.yaml) )

```
! config_otel.yaml X
C: > Source > Repos > dapr-observability > yaml files > otel > ! config_otel.yaml
1  apiVersion: dapr.io/v1alpha1
2  kind: Configuration
3  metadata:
4    name: tracing
5    namespace: default
6  spec:
7    mtls:
8      enabled: false
9    tracing:
10     samplingRate: "1"
11     otel:
12       endpointAddress: "localhost:4317"
13       isSecure: false
14       protocol: grpc
```

**otel-collector-config.yaml:** ( [https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/otel/otel\\_collector\\_config.yaml](https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/otel/otel_collector_config.yaml) )

! otel\_collector\_config.yaml X

C: > Source > Repos > dapr-observability > yaml files > otel > ! otel\_collector\_config.yaml

```
1  # Configure receivers
2  # We only need otlp protocol on grpc, but you can use http, zipkin, jaeger, aws, etc.
3  # https://github.com/open-telemetry/opentelemetry-collector-contrib/tree/main/receiver
4  receivers:
5    otlp:
6      protocols:
7        grpc:
8        http:
9
10 # Configure exporters
11 exporters:
12   #format: proto
13   azuremonitor:
14     endpoint: "https://dc.services.visualstudio.com/v2/track"
15     instrumentation_key: "<<YOUR INSTRUMENTATION KEY>>"
16     # maxbatchsize is the maximum number of items that can be
17     # queued before calling to the configured endpoint
18     maxbatchsize: 100
19     # maxbatchinterval is the maximum time to wait before calling
20     # the configured endpoint.
21     maxbatchinterval: 10s
22
23 # Export to a file
24 file:
25   path: /etc/output/logs.json
26
27 # Configure processors (batch, sampling, filtering, hashing sensitive data, etc.)
28 # https://opentelemetry.io/docs/collector/configuration/#processors
29 processors:
30   batch:
31
32 # Configure pipelines. Pipeline defines a path the data follows in the Collector
33 # starting from reception, then further processing or modification and finally
34 # exiting the Collector via exporters.
35 # https://opentelemetry.io/docs/collector/configuration/#service
36 # https://github.com/open-telemetry/opentelemetry-collector/blob/main/docs/design.md#pipelines
37 service:
38   pipelines:
39     traces:
40       receivers: [otlp]
41       processors: [batch]
42       exporters: [azuremonitor, file]
43     #metrics:
44     #   receivers: [otlp]
45     #   processors: [batch]
46     #   exporters: [logging, prometheus]
47     logs:
48       receivers: [otlp]
```

**docker-compose.yaml :** ( <https://github.com/nsathappan/dapr-observability/blob/main/yaml%20files/otel/docker-compose.yaml> )

```
docker-compose.yaml X
C: > Source > Repos > dapr-observability > yaml files > otel > docker-compose.yaml

1  services:
2    app1Service:
3      image: ${DOCKER_REGISTRY-}app1
4      build:
5        context: .
6        dockerfile: App1/Dockerfile
7      ports:
8        - "40000:41000"
9
10   app1Service-sidecar:
11     image: "daprio/daprd:latest"
12     command: [
13       "./daprd",
14       "-app-id",
15       "app1-Service",
16       "-app-port", "41000",
17       "-components-path", "/components",
18       "--config", "/components/config.yaml",
19       "-log-level", "debug"
20     ]
21     volumes:
22       - "./components:/components"
23     depends_on:
24       - app1Service
25     network_mode: "service:app1Service"
26
27   app2Service:
28     image: ${DOCKER_REGISTRY-}app2
29     build:
30       context: .
31       dockerfile: App2/Dockerfile
32     ports:
33       - "40001:41001"
34
35   app2Service-sidecar:
36     image: "daprio/daprd:latest"
37     command: [
38       "./daprd",
39       "-app-id",
40       "app2-Service",
41       "-app-port", "41001",
42       "-components-path", "/components",
43       "--config", "/components/config.yaml",
44       "-log-level", "debug"
45     ]
46     volumes:
47       - "./components:/components"
48     depends_on:
```

Otel logs:

server-otel-1

425c18ead12 otel/opentelemetry-collector-contrib:latest

4317.4317 8888.8888 Show all ports (3)

STATUS Exited (0) (9 hours ago)

Logs Inspect Bind mounts Exec Files Stats

```
2025-04-02 00:54:31 2025-04-02T04:54:31.506Z info service@v0.122.0/service.go:193 Setting up own telemetry...
2025-04-02 00:54:31 2025-04-02T04:54:31.508Z info service@v0.122.0/service.go:260 Starting otelcol.contrib... ["Version": "0.122.0", "huncPU": 8]
2025-04-02 00:54:31 2025-04-02T04:54:31.508Z info extensions/extensions.go:40 Starting extensions...
2025-04-02 00:54:31 2025-04-02T04:54:31.551Z info otelreceiver@v0.122.0/otlp.go:116 Starting GRPC server ["otelcol.component.id": "otlp", "otelcol.component.kind": "Receiver", "endpoint": "localhost:4317"]
2025-04-02 00:54:31 2025-04-02T04:54:31.552Z info otelreceiver@v0.122.0/otlp.go:173 Starting HTTP server ["otelcol.component.id": "otlp", "otelcol.component.kind": "Receiver", "endpoint": "localhost:4318"]
2025-04-02 00:54:31 2025-04-02T04:54:31.552Z info service@v0.122.0/service.go:283 Everything is ready. Begin running and processing data.

2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.112739532" level=info msg="Starting Dapr Runtime -- version 1.15.3 -- commit c3687714f9592def40b14b390bf5170fe8a1bfff" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.112739532" level=info msg="Log level set to: info" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.112895512" level=warning msg="TLS is disabled. Skipping certificate request and tls validation" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.security type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.1715380752" level=info msg="Enabled features: SchedulerReminders" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.1716887722" level=info msg="metric spec: {\"enabled\":true}" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.diagnostics type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.1717311722" level=info msg="Using default latency distribution buckets: [1 2 3 4 5 6 8 10 13 16 20 25 30 40 50 65 80 100 130 160 200 250 300 400 500 650 800 1000 2000 5000 10000 20000 50000 100000]" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.diagnostics type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.1719587082" level=warning msg="The default value for 'spec.metric.http.increasedCardinality' will change to 'false' in Dapr 1.15 or later" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.diagnostics type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.24213752" level=warning msg="Actor runtime disabled: api error: code = Unavailable desc = placement service is not configured. Actors and Workflow APIs will be unavailable" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.actor type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.2422955972" level=info msg="Actor runtime disabled: api error: code = Unavailable desc = placement service is not configured" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.actor type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.2423139972" level=warning msg="Scheduler disabled, not connecting..." app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.scheduler type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.2423423972" level=info msg="standalone mode configured" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.2423566972" level=info msg="app id: auths-api" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.2424676952" level=info msg="metrics server started on 0.0.0.0:9090/" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.2425276942" level=info msg="Dapr trace sampler initialized: ParentBased(root:AlwaysOnSampler,remoteParentSampled:AlwaysOnSampler,remoteParentNotSampled:AlwaysOffSampler,localParentSampled:AlwaysOnSampler,localParentNotSampled:AlwaysOffSampler)" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.2427144292" level=info msg="Local service entry announced: auths-api -> 172.22.0.6:48228" app_id=auths-api component=hr (ndms/v1)" instance=dd18281e0dfd scope=dapr.contrib type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.2428080992" level=info msg="Initialized name resolution to ndms" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.2429172872" level=info msg="Loading components..." app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:37 ttime="2025-04-02T14:30:37.3439562062" level=info msg="Component loaded: keyvault-secrets-store (secretstores.azure.keyvault/v1)" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.processor type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0494580882" level=info msg="Component loaded: appconfig-store (configuration.azure.appconfig/v1)" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.processor type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0495724272" level=info msg="Component loaded: in-memory-state-store (state.in-memory/v1)" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.processor type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0496312262" level=info msg="Waiting for all outstanding components to be processed." app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0496455382" level=info msg="All outstanding components processed" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0496624442" level=info msg="Loading endpoints." app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0496964862" level=info msg="Waiting for all outstanding http endpoints to be processed." app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0496974492" level=info msg="All outstanding http endpoints processed" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.049698942" level=info msg="Loading declarative Subscriptions." app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0496989542" level=info msg="gRPC server listening on TCP address: :50001" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.grpc type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0496991562" level=info msg="Enabled gRPC tracing middleware" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.grpc type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497041642" level=info msg="Enabled gRPC metrics middleware" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.grpc type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.049705782" level=info msg="Registering workflow engine for gRPC endpoint: [::]:50001" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.grpc type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497078192" level=info msg="API gRPC server is running on port 50001" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497303982" level=warning msg="The default value for 'spec.metric.http.increasedCardinality' will change to 'false' in Dapr 1.15 or later" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.http type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497328662" level=info msg="Enabled max body size HTTP middleware with size 4194304 bytes" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.http type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497399072" level=info msg="Enabled tracing HTTP middleware" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.http type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497412882" level=info msg="Enabled metrics HTTP middleware" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.http type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497464652" level=info msg="HTTP server listening on TCP address: :3500" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.http type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497477742" level=info msg="HTTP server is running on port 3500" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497496762" level=info msg="The request body size parameter is: 4194304 bytes" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497519812" level=info msg="gRPC server listening on TCP address: :48228" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.grpc internal type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497529382" level=info msg="Enabled gRPC tracing middleware" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.grpc internal type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497531842" level=info msg="Enabled gRPC metrics middleware" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime.grpc internal type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.04975477912" level=info msg="Internal gRPC server is running on :48228" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.04975611992" level=info msg="actors: state store is not configured - this is okay for clients but services with hosted actors will fail to initialize" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:38 ttime="2025-04-02T14:30:38.0497598012" level=info msg="application protocol: http, waiting on port 31000. This will block until the app is listening on that port." app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:40 ttime="2025-04-02T14:30:40.1449298232" level=info msg="Application discovered on port 31000" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:40 ttime="2025-04-02T14:30:40.348265162" level=info msg="Application configuration loaded" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:40 ttime="2025-04-02T14:30:40.3483064282" level=info msg="Dapr initialized. Status: Running. Init Elapsed 3109ms" app_id=auths-api instance=dd18281e0dfd scope=dapr.runtime type=log ver=1.15.3
2025-04-02 10:30:52 2025/04/02 14:30:52 traces export: context deadline exceeded: rpc error: code = Unavailable desc = connection error: desc = "transport: Error while dialing: dial tcp 127.0.0.1:4317: connect: connection refused"
```

**Issue 2: Logs** We aim to export all application logs to AppInsights for centralized management. Currently, logs are visible only in stdout locally. We attempted to use the same configuration as *Option 2* (OpenTelemetry Collector) above but were unsuccessful.



