**Description of OpenTelemetry Collector Configuration and Docker Monitoring Issue**

The YAML config is for an OpenTelemetry Collector, as part of a SigNoz deployment in github.com/rhildred/signoz repo, to receive, process, and forward telemetry data (logs, metrics, traces) for observability. SigNoz is an open-source tool that employs OpenTelemetry for application monitoring, including Docker containers. The setup defines receivers to read data, processors to transform it, exporters to send it to a ClickHouse database, and service pipelines to route data. However, the Docker monitoring feature, which is meant to watch container metrics (e.g., CPU, memory), is not working. This answer outlines the configuration, makes informed speculations on the failure from the provided otel-collector-config.yaml and web research, derives two hypotheses for the failure, and details testing in two repository branches, satisfying the Level 4 criterion for PROG8850 by being understandable and operational.

Configuration Overview:

• Receivers:

• tcplog/docker: Listens on port 2255 for Docker container logs, parsing them with a regex to forward fields like timestamp, container\_id, container\_name, and body.

•\totlp, jaeger: Accept traces and metrics via OTLP (ports 4317, 4318) and Jaeger protocols (14250, 14268).

•\thostmetrics: Scrapes system metrics (CPU, memory, disk) every 30 seconds.

•\tprometheus: Scrapes collector metrics from localhost:8888.

•\tProcessors:

•\tbatch: Batches data (10,000 items, up to 11,000) for exporting efficiently.

•\tsignozspanmetrics: Generates latency metrics from traces for ClickHouse.

•\tresourcedetection: Adds metadata (e.g., host.name, k8s.cluster.name).

•\tExporters: Export data into ClickHouse tables (signoz\_traces, signoz\_metrics, signoz\_logs) at tcp://clickhouse:9000.

• Service Pipelines: Forward route traces, metrics, and logs through specified receivers, processors, and exporters, with health check and profiling extensions.

Problem: The repo's Docker monitoring, which is presumably intended to show container metrics in the SigNoz UI (http://localhost:3301), is not functional. The tcplog/docker receiver collects logs, but there is no docker\_stats receiver for metrics, a required component based on SigNoz documentation.

**Hypotheses for Failure:**

**1. Absent docker\_stats Receiver:**Problem: No docker\_stats receiver, required for collecting Docker container metrics (e.g., container\_cpu\_utilization). Only tcplog/docker receiver is being used for logs.

• Repair: Add docker\_stats with endpoint: "unix:///var/run/docker.sock" and collection\_interval: 10s, and append it to the metrics pipeline. For compatibility reasons, use signoz/signoz-otel-collector:0.111.24.

• Test: In branch fix-docker-stats-receiver, update otel-collector-config.yaml and docker-compose.yaml, run docker-compose up -d, and check SigNoz UI for metrics. Logs (docker logs signoz-otel-collector) reveal errors if there are any missing metrics.

**2. Docker Socket Access Issue:**

• Problem: The receiver docker\_stats needs access to /var/run/docker.sock. If Docker Compose fails to mount the socket or permissions are not properly configured, metric collection will be a failure.

• Fix: Mount /var/run/docker.sock and /var/lib/docker/containers in docker-compose.yaml, set user: root for permission, and include resourcedetection/docker processor for metadata.

• Test: In branch fix-docker-socket, apply changes, run docker-compose up -d, and verify metrics in SigNoz UI. Confirm logs for "permission denied" exceptions.

Testing Plan:

• Fork Repository: Replicate forking github.com/rhildred/signoz to github.com/your-username/signoz.

• Branches:

• fix-docker-stats-receiver: Add docker\_stats receiver and pipeline update.

•.fix-docker-socket: Configure socket access and add resourcedetection/docker.

• Updates: Update otel-collector-config.yaml and docker-compose.yaml (artifacts above). Update README.md for each branch with hypothesis, changes, and test results.

• Expected Results:

• Hypothesis 1: Metrics are present in SigNoz UI if docker\_stats is the only problem. Failure shows socket access or ClickHouse problems.

•\tHypothesis 2: Docker metadata statistics appear if socket access was the issue. If it fails, there are deeper permission or version issues.