

# Pattern: Log aggregation

## Context

You have applied the Microservice architecture pattern ([../microservices.html](#)). The application consists of multiple services and service instances that are running on multiple machines. Requests often span multiple service instances.

Each service instance generates writes information about what it is doing to a log file in a standardized format. The log file contains errors, warnings, information and debug messages.

## Problem

How to understand the behavior of an application and troubleshoot problems?

## Forces

- Any solution should have minimal runtime overhead

## Solution

Use a centralized logging service that aggregates logs from each service instance. The users can search and analyze the logs. They can configure alerts that are triggered when certain messages appear in the logs.

## Examples

- AWS Cloud Watch (<https://aws.amazon.com/cloudwatch/>)

## Resulting Context

This pattern has the following issues:

- handling a large volume of logs requires substantial infrastructure

## Related patterns

- Distributed tracing ([distributed-tracing.html](#)) - include the external request id in each log message
- Exception tracking ([exception-tracking.html](#)) - as well as logging exceptions, report them to an exception tracking service

---

[Tweet](#)

[Follow @MicroSvcArch](#)