Design for operations

08/30/2018 • 2 minutes to read • Contributors

In this article

Design an application so that the operations team has the tools they need

The cloud has dramatically changed the role of the operations team. They are no longer responsible for managing the hardware and infrastructure that hosts the application. That said, operations is still a critical part of running a successful cloud application. Some of the important functions of the operations team include:

- Deployment
- Monitoring
- Escalation
- Incident response
- Security auditing

Robust logging and tracing are particularly important in cloud applications. Involve the operations team in design and planning, to ensure the application gives them the data and insight thay need to be successful.

Recommendations

Make all things observable. Once a solution is deployed and running, logs and traces are your primary insight into the system. *Tracing* records a path through the system, and is useful to pinpoint bottlenecks, performance issues, and failure points. *Logging* captures individual events such as application state changes, errors, and exceptions. Log in production, or else you lose insight at the very times when you need it the most.

Instrument for monitoring. Monitoring gives insight into how well (or poorly) an application is performing, in terms of availability, performance, and system health. For example, monitoring tells you whether you are meeting your SLA. Monitoring happens during the normal operation of the system. It should be as close to real-time as possible, so that the operations staff can react to issues quickly. Ideally, monitoring can help avert problems before they lead to a critical failure. For more information, see Monitoring and diagnostics.

Instrument for root cause analysis. Root cause analysis is the process of finding the underlying cause of failures. It occurs after a failure has already happened.

Use distributed tracing. Use a distributed tracing system that is designed for concurrency, asynchrony, and cloud scale. Traces should include a correlation ID that flows across service boundaries. A single operation may involve calls to multiple application services. If an operation fails, the correlation ID helps to pinpoint the cause of the failure.

Standardize logs and metrics. The operations team will need to aggregate logs from across the various services in your solution. If every service uses its own logging format, it becomes difficult or impossible to get useful information from them. Define a common schema that includes fields such as correlation ID, event name, IP address of the sender, and so forth. Individual services can derive custom schemas that inherit the base schema, and contain additional fields.

Automate management tasks, including provisioning, deployment, and monitoring. Automating a task makes it repeatable and less prone to human errors.

Treat configuration as code. Check configuration files into a version control system, so that you can track and version your changes, and roll back if needed.