Deployment and Operations for Software Engineers 2nd Ed

Chapter 13—Post production

Outline

Testing in production

Service level thresholds

Incident response

Testing in Production

- There are two different types of testing that can happen after your service has been deployed.
  - Live testing where known problems are introduced into your system and the response measured. Chaos Engineering is a term for this type of testing.
  - Passive testing where scanners look for various types of problems and, may, perform remediation action.

Chaos Engineering

- Chaos engineering is defined as “the discipline of experimenting on a distributed system in order to build confidence in the system’s capability to withstand turbulent conditions in production.”
- An experiment (or test) introduces a failure into the system. There is a hypothesis about what should happen, which defines the passing conditions for the test
- Before testing in the production environment, your organization needs to have sufficient confidence in its infrastructure and systems that few tests will fail and that the test failures will not have catastrophic consequences.

Chaos Engineering tools

- Two well known tools to introduce failures are
  - Chaos Monkey–randomly shuts down virtual machines in production to ensure that small disruptions will not affect the overall service.
  - Latency Monkey–simulates a degradation of network service and checks to make sure that upstream services react appropriately.

Environment scanners

- Scanning a production system is much less disruptive than Chaos Engineering.
- A scanning tool examines the production environment looking for different types of problems.
- A scanning tool uses resources but does not perturb the running system except if specific problems are found.

Scanning tools

- *Conformity scanners*–detects instances that aren't coded to best practices and shuts them down, giving the service owner the opportunity to relaunch them properly. Best practices here include phenomena that are externally visible such as generating logs.
- *Security scanners*–searches out security weaknesses. It may terminate the offending instances. It also ensures that SSL and DRM (Digital Rights Management) certificates are not expired or close to expiration. A security scanner can also check whether any new vulnerabilities have been found for the components in your production system.
- *Health scanners*–performs health checks on each instance and monitors other external signs of process health such as CPU and memory usage.
- *Janitor scanner*–searches for unused resources and discards them. .

Discussion questions

1. How does a janitor scanner work?
2. How does a security scanner work?

Outline

Testing in production

Service level thresholds

Incident response

Service level agreement

- An Service Level Agreement (SLA)
  - Is an agreement between a service and its clients
  - identifies a metric (e.g., request latency for a service) and a threshold (e.g., 99% of requests will receive a response within 300 milliseconds).
- A typical agreement will contain many of these metric/threshold requirements.
- If the client is outside your organization, these SLAs may be part of a legal contract.

Service Level Objective

- A Service Level Objective (SLO) is an alerting value for an SLA.
- It is designed to generate an alert and is set to a value that gives you time to prevent a violation of an SLA from occurring.

Service Level Indicator

- Some SLOs are not directly observable. Availability, for example, is a common SLA and, hence, SLO.
- Availability, however, cannot be directly measured and so you define a surrogate. This is the Service Level Indicator (SLI).
- The SLI for availability is typically error rates for requests. Thresholds are set in terms of SLIs.

Monitoring SLIs

- The SLIs are monitored on some periodic basis. The period will depend on the maturity of your service (shorter when the service is newly installed), and the criticality of your service to the overall meeting of the SLA.
- Instrumentation is providing measurements–placing an instrument in your software.
- Telemetry is automatically sending the data gathered by an instrument to a recording site.

Typical SLIs

- *Latency and throughput*. The time between a message arriving at a service and the response being returned. Recording time of arrival and time of response allows the calculation of both latency and throughput.
- *Request satisfaction rate.* Recording a request on arrival and whether it was satisfactorily served on response is an availability measure.
- *Traffic*. Number of requests arriving at your service per unit time.
- *Saturation*. The measure of utilization of the resources (CPU, network, memory) that your service relies on.

Discussion questions

1. What is a quality in addition to availability that cannot be directly measured? What is an SLI for that quality?
2. What does an availability of 99.99% actually mean?

Outline

Testing in production

Service level thresholds

Incident response

Incidents

- An Incident an event that could lead to loss of, or disruption to, an organization's operations, services or functions.
- In software terms, an incident is either a performance or availability problem.
- Security incidents are dealt with by security specialists who may be a portion of a response team.

Life cycle of an incident

- An incident is detected by a monitoring system based on the threshold values you have established.
- The monitoring system then sends out a page to a first responder.
- The monitoring system also enters the incident in an incident repository. The incident repository is frequently called a *ticketing system*.

Life cycle of an incident

- There are two goals when an incident occurs.
  - The immediate goal is to fix the current problem. Get the system running normally again.
  - The second goal is to determine the root cause of the incident so that it does not happen again.
- The first responder (possibly with assistance from others) diagnoses the cause of the incident and determines a quick fix or work around.
- Once the system is working normally, the ticket is closed.

Post incident activities

- The first responder, in conjunction with the development team, determines the root cause of the incident.
- This is entered into the ticketing system.
- It is the responsibility of the development team to act on the root cause.

Ensuring qualities

- Three models for ensuring quality in systems.
  - “You Build it, You run it” due to Amazon
  - Site Reliability Engineers due to Google
  - Production Engineering due to Meta (Facebook)

You Build it, You Run it

- *There is another lesson here: Giving developers operational responsibilities has greatly enhanced the quality of the services, both from a customer and a technology point of view. The traditional model is that you take your software to the wall that separates development and operations and throw it over and then forget about it. Not at Amazon. You build it, you run it. This brings developers into contact with the day-to-day operation of their software. It also brings them into day-to-day contact with the customer. This customer feedback loop is essential for improving the quality of the service.*

  -Werner Vogels

- Werner Vogels was the Amazon CTO as it grew into the ubiquitous online platform that it is today.

Amazon model

- Developers are responsible for post-production operations.
- This includes having developers carry pagers and be the first responders in the event of an operational problem.
- Pager duty is rotated among team members so that no single person is always on call.

Analyzing the Amazon model

- Assumptions. A developer
  - is familiar with the internals of the service,
  - can interpret the information collected,
  - provides edback as to modifications to deal with the root cause of a problem that will be heeded by fellow team members.
- Drawbacks
  - This model assumes that because a service triggered an alert, the service has a problem.
  - The problem can stem from upstream clients or downstream dependencies.
  - The problem may be in the network.
  - A developer of a service may or may not have a good understanding of the environment in which their service operates.

Site Reliability Engineer

- Google, Netflix, and now many other organizations have a separate team called Site Reliability Engineers (SREs).
- A system is assigned to a SRE team,
- A member of that team is a first responder when an incident occurs.
- Pager duty is rotated among members of the team.

The SRE model

- SREs have an overall understanding of a system and its environment.
  - Their knowledge is broader than that of a single service developer.
  - It is not as deep with any individual service.
- SREs oversee monitoring and SLIs.
- They also share a knowledge base of problems with the network, with systems, and with techniques for trouble shooting problems.
- To encourage development teams to listen to the advice given them by SRE teams, an SRE team has the option to refuse to support particular systems.

Production Engineering

- Production Engineering is Meta's (Facebook) philosophy for ensuring quality systems.
- A production engineer is responsible for reliability, scalability, performance, and security for production services.
- They are a separate organizational unit (like SREs) but are embedded into development teams (like you build it, you run it).
- Their skill set is similar to SREs in that they must have a broad understanding of the infrastructure and its components.
- Since they are embedded in development teams, they acquire a detailed knowledge of the internals of particular services.

Discussion questions

1. What is your reaction to your wearing a pager and being on call 24/7 for a week?
2. Why would a development team not act on the recommendations of the first responders?