Carnegie Mellon

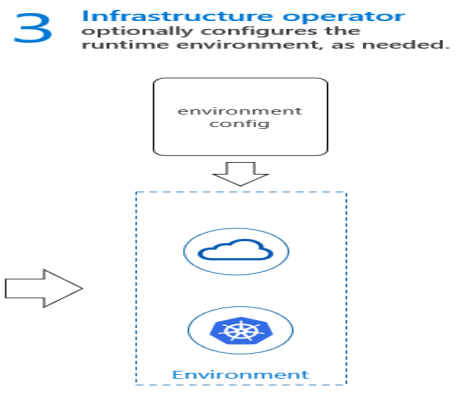
DevOps: Engineering for Deployment and Operations

Postproduction

1

Carnegie Mellon University

Open Application Model



institute for
SOFTWARE
RESEARCH

2

Carnegie Mellon University

Key concepts

- Incident - an event that could lead to loss of, or disruption to, an organization's operations, services or functions.
  - May be minor, such as running out of disk space
  - May be major , such as data breach
- Telemetry – collection of information for monitoring environmental conditions

Carnegie Mellon University

Overview

- **Telemetry**
- Incident response
- Live testing

Carnegie Mellon University

Scenario

- It is 3:00AM and your pager goes off.
- There is a problem with your service!
- You get out of bed and log onto the production environment and look at the services dashboard.
- One instance of your service has high latency
- You drill down and discover the problem is a slow disk
- You move temporary files for your service to another disk and place the message “replace disk” on the operators queue.

Carnegie Mellon University

Troubleshooting process

- First step is to isolate problem
  - Current service
  - Upstream service (too many requests)
  - Downstream service (too slow)
- Second step is to decide whether it is a hardware or software problem
  - What has changed in the software?
  - Has hardware shown signs of problems with other services?
  - If a single instance of multiple instances has problems, look for hardware first.

Carnegie Mellon University

Single service – single server

- Look at following data
  - CPU
  - Memory
  - I/O activity
  - Number of requests
  - Response time to inbound requests
  - Response time for outbound requests
  - Error rates
- Look for abnormal values

Carnegie Mellon University

Single service – multiple servers

- Multiple servers served through a load balancer
- Look at same set of data as for single server
  - CPU
  - Memory
  - I/O activity
  - Number of requests
  - Response time to inbound requests
  - Response time for outbound requests
  - Error rates
- Look at aggregate values over multiple servers

Carnegie Mellon University

Isolating problem

- Is problem with this service or client or dependent services?
- If problem is with this service is it manifested across all servers or just one. I.e. drill down into aggregates to get individual values

© Len Bass 2015

institute for SOFTWARE RESEARCH

Carnegie Mellon University

Multiple services – multiple servers

- Same basic strategy
  - Isolate problem through identifying problem by looking at aggregates
  - Drill down to decide service and server that contributes to problem
  - Look at what has changed in software and whether hardware has manifested problems earlier

© Len Bass 2015

institute for SOFTWARE RESEARCH

Carnegie Mellon University

Overall requirements from this sequence of trouble shooting

- Gather variety of different kinds of data
  - Either resource usage or things that contribute to resource usage
  - Ensure each data item can be traced as to source and activity
- Collect data into a location where it can be queried and drilled into.

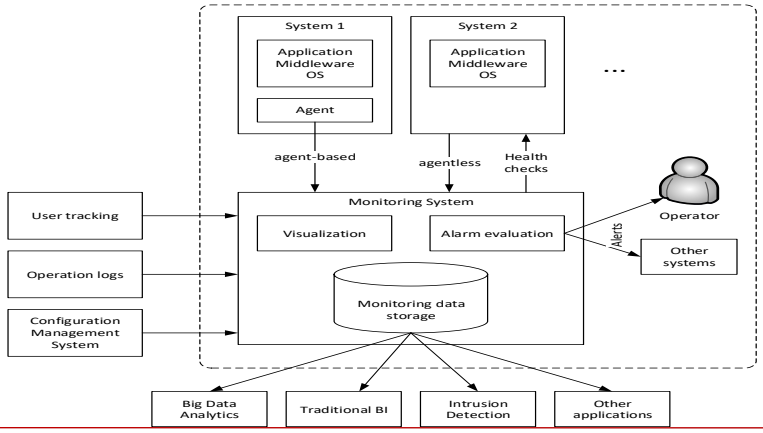
Carnegie Mellon University

Information needs

- Metrics collected by infrastructure
- Logs from instance with relevant information
- Central repository for logs
- Dashboard that displays metrics
- Alerting system
  - Monitoring latency of instances
  - Rule: if high latency then alarm

Carnegie Mellon University

Architecture of Monitoring System



Carnegie Mellon University

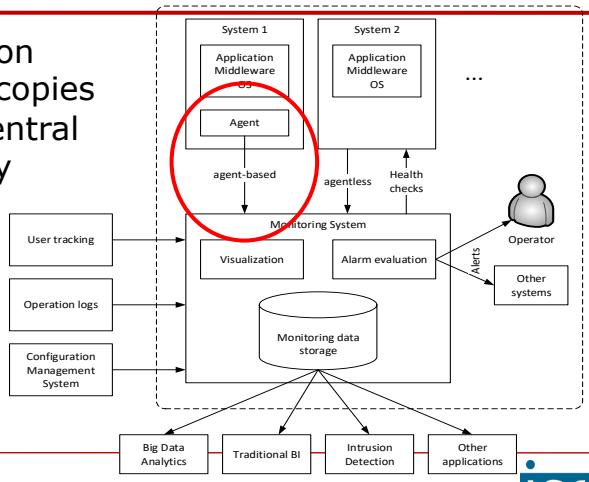
Logs

- A log is an append only data structure
- Written by each software system.
- Located in a fixed directory within the operating system
- Enumerates events from within software system
  - Entry/exit
  - Troubleshooting
  - DB modifications
  - …

Carnegie Mellon University

Instance Log

Daemon on instance copies logs to central repository



Carnegie Mellon University

Logs on Entry/Exit

- Recall that Protocol Buffers automatically generate procedures that are called on entry/exit to a service
- These procedures can be made to call logging service with parameters and identification information.
- Logs on entry/exit can be made without additional developer activity

Carnegie Mellon University

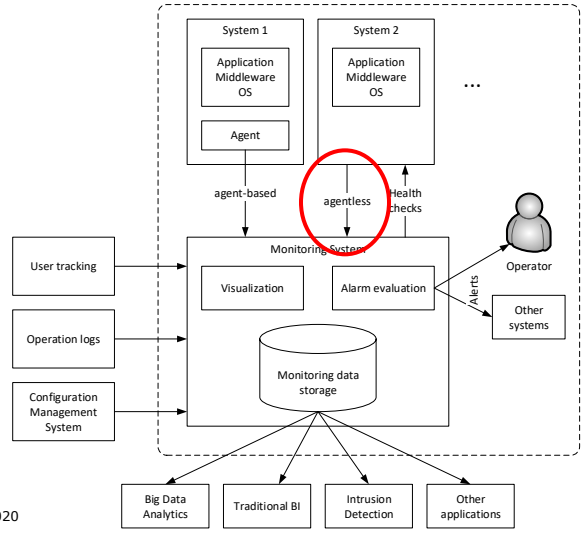
Metrics

- Metrics are measures of activity over some period of time
- Collected automatically by infrastructure over externally visible activities of VM
  - CPU
  - I/O
  - etc

institute for SOFTWARE RESEARCH

Carnegie Mellon University

Metrics collected by infrastructure



institute for SOFTWARE RESEARCH

Carnegie Mellon University

Repository

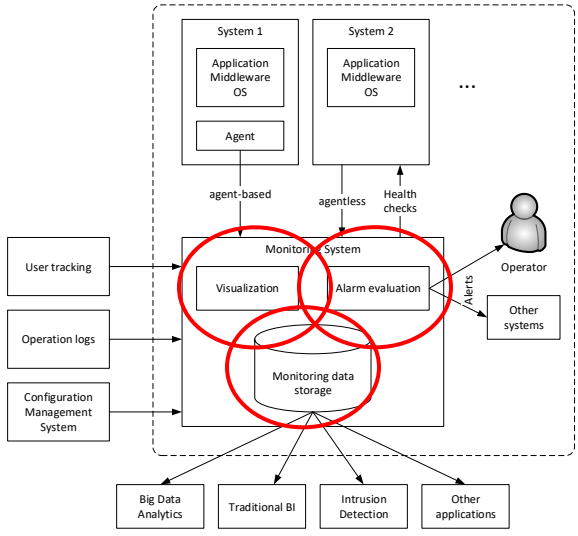
- Logs and metrics are placed in central repository
- Repository generates alarms based on rules
- Provides central location for examination when problem occurs
- Displays information in dashboard that allows for drilling down to understand source of particular readings.

institute for SOFTWARE RESEARCH

19

Carnegie Mellon University

Central Repository with alerting and dashboard



institute for SOFTWARE RESEARCH

20

Carnegie Mellon University

Overview

- Telemetry
- **Incident response**
- Live testing

institute for SOFTWARE RESEARCH

Carnegie Mellon University

Incident response

- Incident occurs
  - can be a result of telemetry data or externally caused.
- Incident response is the managing of the aftermath of the incident.
- Ideal response:
  - Restore the system to production
  - Analyze cause of incident
  - Prevent the incident from re-occurring

institute for SOFTWARE RESEARCH

Carnegie Mellon University

Two incident response philosophies

- You build it, you run it (originated by Amazon)
- Site Reliability Engineers (SRE) (originated by Google)

Carnegie Mellon University

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."

-Wener Vogels

https://queue.acm.org/detail.cfm?id=1142065

Carnegie Mellon University

SRE

- Separate organizational unit whose responsibility is to manage incidents.
- Coordination enables detection of system outage patterns
- SRE team rotates pager duty
- Term for an SRE is ~2-3 years. High stress and they burn out. Ex SREers go back to production unit.

Carnegie Mellon University

SRE mindset

- "Here's what you do when someone breaks something or finds something very difficult to debug: You say thank you. Thank you for finding this edge case. Thank you for highlighting this overcomplicated part of our system. Thank you for pointing out this gap in our docs. And then you go make it so nobody can break it the same way again."
- Tanya Reilly https://landing.google.com/sre/

Carnegie Mellon University

Overview

- Telemetry
- Incident response
- **Live testing**

institute for SOFTWARE RESEARCH

Carnegie Mellon University

Live testing

- Netflix has a “Simian Army” to perform testing after a service is in production.
  - Chaos Monkey kills production processes
  - Latency Monkey introduces extra latency into the network.
  - Various other monkeys perform janitor services
    - Looking for certificates or licenses about to expire
    - Ensuring appropriate localization
    - Cleaning up unused resources
    - Ensuring security groups are appropriately used.

institute for SOFTWARE RESEARCH

Carnegie Mellon University

Summary

- Developers may carry pagers and be first responders
- Determining problem requires access to a wide variety of data
  - Logs
  - Metrics
- Postproduction testing may introduce errors or provide janitorial services

institute for SOFTWARE RESEARCH