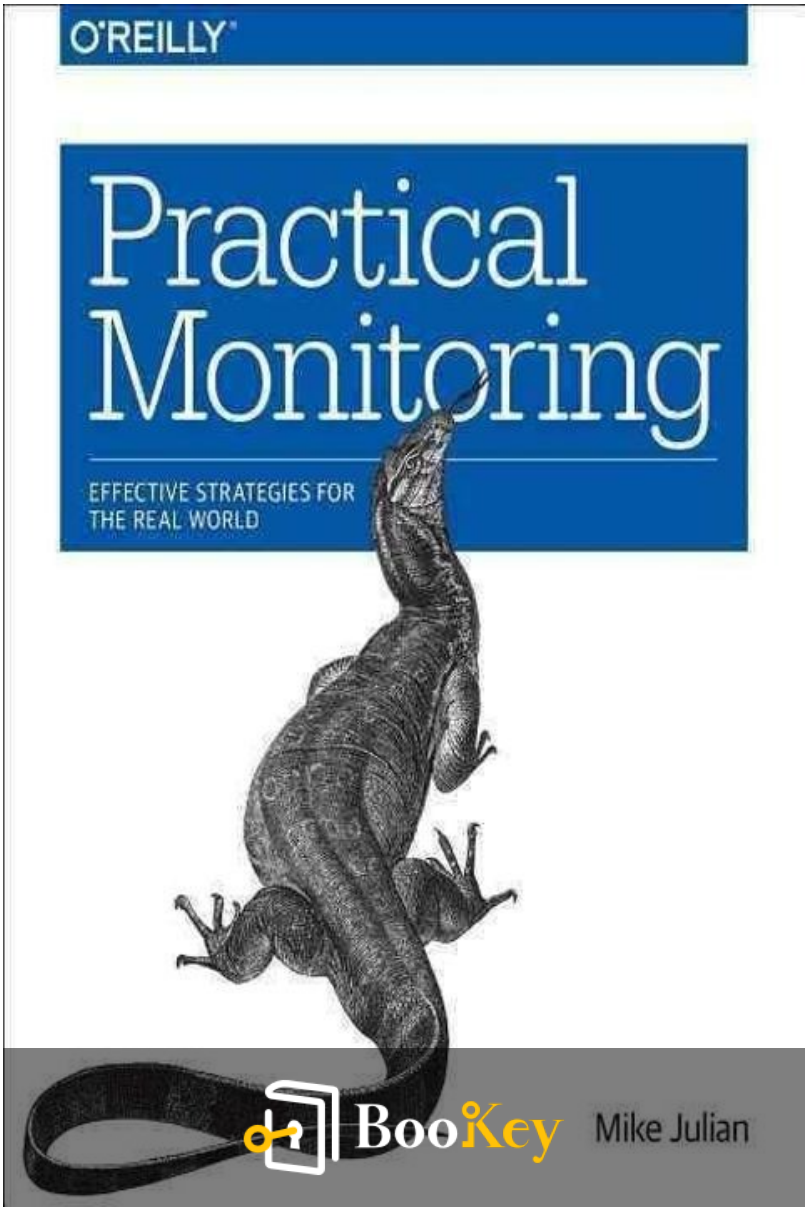
Practical Monitoring PDF

Mike Julian



Practical Monitoring

Master Your Monitoring Strategy for Today's Complex Infrastructure

Written by Bookey

About the book

Are you seeking to enhance your monitoring practices but unsure where to start? "Practical Monitoring" is your essential guide. This book delves into the elements that may be hindering your monitoring effectiveness and offers a pragmatic framework for developing and executing a comprehensive monitoring strategy, encompassing everything from applications to datacenter hardware. In an era marked by microservices and cloud infrastructure, monitoring is more vital than ever. This resource introduces new methodologies, innovative tools, and a foundational understanding of statistics and monitoring terminology. Designed for operations engineers, system administrators, site reliability engineers, and network professionals, it empowers you to pinpoint critical monitoring areas and develop effective approaches to optimize your systems.

About the author

Mike Julian is a seasoned monitoring expert and consultant with a wealth of experience in the fields of system performance and operational excellence. With a professional background that spans many years in the tech industry, he has honed his skills in designing and implementing effective monitoring systems for complex infrastructure environments. As the founder of the online training platform "Practical Monitoring," Mike is dedicated to sharing his insights and best practices with professionals looking to enhance their monitoring capabilities. His expertise is not just theoretical; he has been at the forefront of helping organizations navigate the intricacies of monitoring in today’s fast-paced digital landscape, making him a respected voice in the community. Through his work, Mike Julian emphasizes the critical role of monitoring in achieving reliability and operational success, firmly establishing himself as an authoritative figure in the realm of practical monitoring solutions.

Summary Content List

Chapter 1 : 1. Monitoring Anti-Patterns

Chapter 2 : 2. Monitoring Design Patterns

Chapter 3 : 3. Alerts, On-Call, and Incident Management

Chapter 4 : 4. Statistics Primer

Chapter 5 : 5. Monitoring the Business

Chapter 6 : 6. Frontend Monitoring

Chapter 7 : 7. Application Monitoring

Chapter 8 : 8. Server Monitoring

Chapter 9 : 9. Network Monitoring

Chapter 10 : 10. Security Monitoring

Chapter 11 : 11. Conducting a Monitoring Assessment

Chapter 12 : A. An Example Runbook: Demo App

Chapter 13 : B. Availability Chart

Chapter 1 Summary : 1. Monitoring Anti-Patterns

Chapter 1: Monitoring Anti-Patterns

Introduction

Before embarking on effective monitoring practices, it's crucial to recognize and rectify bad habits, also known as anti-patterns. These habits often begin with good intentions but evolve into detrimental practices due to a lack of adequate tools, legacy systems, and fear of change.

Anti-Pattern #1: Tool Obsession

A common issue is an excessive focus on monitoring tools rather than capabilities. Organizations often fall into the trap of believing that acquiring new tools will solve their monitoring problems. However, monitoring is a complex issue that requires multiple tools and approaches customized to specific needs. The myth of a "single-pane-of-glass" tool is prevalent but misleading; effective monitoring requires diverse and specialized tools tailored to various aspects of the system.

Monitoring Is Multiple Complex Problems Under One Name

Monitoring encompasses various complex challenges that cannot be resolved by a single tool. A toolbox of general and specialized tools is essential, tailored to the unique needs of the environment.

The Observer Effect Isn’t a Problem

Concerns about the observer effect impacting system performance are largely unfounded. Modern systems can handle the load of monitoring tools without significant

issues. Agents for metrics collection are advised over agentless monitoring for better visibility and control.

How Many Tools Is Too Many?

There’s no definitive answer to the number of monitoring tools one should use. The focus should be on the job they accomplish rather than quantity. Consolidation is beneficial when tools overlap in capability, but having multiple tools servicing different needs is acceptable as long as they work harmoniously.

Avoid Cargo-Culting Tools

Tools and procedures adopted from successful teams may not yield the same results if their underlying norms do not align with a different team's culture. Evaluate and adapt solutions instead of copying them blindly.

Sometimes, You Really Do Have to Build It

Creating specialized tools tailored to specific problems can be advantageous. Custom solutions may be necessary when existing tools do not meet particular needs.

The Single Pane of Glass Is a Myth

The desire for a singular view of monitoring results is unrealistic. Successful monitoring typically involves multiple tools and dashboards to effectively capture complex data.

Anti-Pattern #2: Monitoring-as-a-Job

Monitoring shouldn't be relegated to a single person or team—it should be a shared skill. Every team member should participate in the monitoring process to ensure a comprehensive understanding and better overall system reliability.

Anti-Pattern #3: Checkbox Monitoring

Checkbox monitoring involves implementing monitoring systems merely to satisfy requirements, resulting in ineffective monitoring. Effective monitoring should be proactive and focused on understanding what "working" means for various systems.

What Does “Working” Actually Mean?

Establish high-level checks in collaboration with service owners to ascertain service functionality, rather than relying solely on metrics like CPU or memory usage.

OS Metrics Aren’t Very Useful for Alerting

Alerting should focus on service performance rather than low-level metrics. Operating system metrics are valuable for diagnostics, but they shouldn't be the basis of alerts without context regarding their operational status.

Collect Your Metrics More Often

Increasing the frequency of metric collection can reveal critical issues that hourly or less frequent checks may miss.

Anti-Pattern #4: Using Monitoring as a Crutch

Relying on monitoring tools to compensate for poorly designed applications is misguided. The focus should shift to stabilizing the service rather than just adding monitoring to mask underlying problems.

Anti-Pattern #5: Manual Configuration

Automation in monitoring configuration is vital. Manual setups waste time and can lead to inadequate monitoring coverage.

Monitoring Cloud Architectures Versus Traditional Ones

Cloud monitoring requires an aggregate approach rather than monitoring individual components. Automation should be ingrained in the monitoring strategy for efficiency and effectiveness.

Wrap-Up

The chapter identifies five critical anti-patterns in monitoring practices: tool obsession, monitoring being seen as a job, checkbox monitoring, misuse of monitoring as a crutch, and manual configuration. Addressing these anti-patterns lays the groundwork for building effective monitoring habits.

Example

Key Point:Identifying Anti-Patterns in Monitoring Practices

Example:As you set up monitoring for your new application, you may feel tempted to buy the latest shiny tool that promises to solve all your problems. However, instead of solely investing in one tool, it’s essential to evaluate your specific needs and create a diverse toolkit. For example, imagine your team is dealing with an influx of user complaints and slow response times. By analyzing your environment, you find that a combination of log analyzers, performance monitors, and alerts for specific service thresholds could better address the situation than relying on a single monitoring solution. This comprehensive approach not only prevents falling into the trap of tool obsession but also enhances your ability to identify and resolve issues promptly.

Critical Thinking

Key Point:Tool obsession can hinder effective monitoring practices.

Critical Interpretation:The chapter suggests that an excessive obsession with acquiring tools can lead organizations away from focusing on the actual monitoring capabilities they need. While the author argues that diverse and tailored tools are essential for successful monitoring, it warrants skepticism. Organizations might benefit from evaluating the effectiveness of their existing tools rather than continually adding new ones, as this could lead to complexity and inefficiency in monitoring practices. As highlighted in 'The Phoenix Project' by Gene Kim, the best solutions often come from optimizing existing processes and tools, rather than merely stacking new ones on top.

Chapter 2 Summary : 2. Monitoring Design Patterns

Chapter 2: Monitoring Design Patterns

In this chapter, Mike Julian discusses four essential design patterns for effective monitoring systems, moving beyond common pitfalls encountered in monitoring practices.

Pattern #1: Composable Monitoring

Composable monitoring promotes the idea of utilizing multiple specialized tools and integrating them loosely together to form a flexible monitoring platform, contrasting with the older monolithic designs. This approach encourages

using best-of-breed tools for different aspects of monitoring, promoting ease of replacement and scalability. The five critical components of a monitoring service include:
- Data Collection
- Data Storage
- Visualization
- Analytics and Reporting
- Alerting

Data Collection

Two methodologies for data collection are highlighted: push and pull models. While the pull model involves a centralized service requesting data, the push model has clients sending data autonomously. Each method has its use cases, with pros and cons, including scalability challenges in centralized systems.

Data Storage

Metrics, often stored in Time Series Databases (TSDB), and logs, which can be structured or unstructured, need to be stored efficiently. The chapter explains the nuances of metrics (counters and gauges) and logs (structured vs.

unstructured) while emphasizing their importance in monitoring.

Visualization

Good visualization of monitoring data through charts and dashboards is essential. Users appreciate customizable dashboards that present relevant data clearly, unlike rigid monolithic tools. The effectiveness of visualization techniques is crucial for operational efficiency.

Analytics and Reporting

SLAs (Service Level Agreements) tie closely into reporting, where data accuracy enables effective service evaluations. The calculation of availability percentages serves as a key performance metric, though it's noted that such metrics must be treated with caution due to potential sampling errors.

Pattern #2: Monitor from the User Perspective

Julian emphasizes starting monitoring efforts close to the user’s interaction points. This approach ensures that monitoring reflects user experiences and system performance

from the user's viewpoint, guiding the monitoring focus on indicators like HTTP response codes and request latency.

Pattern #3: Buy, Not Build

The author advocates for utilizing ready-made SaaS solutions for monitoring as opposed to building in-house systems, especially when monitoring needs are nascent. Focusing on established SaaS tools allows for quicker implementation and the ability to leverage expertise that may not be available internally.

Pattern #4: Continual Improvement

Julian stresses the need for incremental progress over time in refining monitoring systems. Companies should expect to evolve their monitoring tools and practices continually and not rush to achieve advanced monitoring levels.

Wrap-Up

Through the understanding and application of these four patterns, organizations can significantly enhance their monitoring systems. The chapter concludes by emphasizing

the importance of moving forward with these principles before diving deeper into specific areas like alert design.

Example

Key Point:Composable Monitoring as a Design Pattern

Example:Imagine you're a system administrator who struggles with inefficient, outdated monitoring tools that don't integrate well. Instead of tackling this problem with a single, bulky solution, picture yourself assembling the best tools for different monitoring needs, like a sculptor choosing the finest chisel for each detail. You select a specialized data collector that pushes metrics seamlessly, store logs in a flexible database, and create stunning, customizable dashboards that reflect real-time performance. Each piece fits into your monitoring platform effortlessly, and if one tool doesn't meet your evolving demands, you simply replace it. This composable approach not only streamlines your operations but also allows for greater scalability, as each part can grow independently while still being part of a cohesive whole.

Chapter 3 Summary : 3. Alerts, On-Call, and Incident Management

Section	Summary
Importance of Alerts	Alerts are crucial for monitoring systems, particularly during unexpected infrastructure issues, and should focus on understanding system behavior over time.
Challenges of Effective Alerting	Effective alerting faces challenges such as false alarms from variable metrics and the limits of human attention.
Characteristics of a Good Alert	A good alert should create urgency and clearly differentiate between actionable alerts and informational notifications.
Key Practices for Effective Alerts	Stop using Email for Alerts Write Runbooks for remediation context Avoid Rigid Thresholds; use dynamic ones Delete and Tune Alerts regularly Use Maintenance Periods to silence alerts Attempt Automated Self-Healing first
On-Call Considerations	Reduce burnout from on-call duties by tuning alerts and fostering a non-blame culture for learning and improvement.
Building a Better On-Call Rotation	Structured on-call rotations can manage workload and improve experience, including follow-the-sun models for continuous coverage.
Incident Management	A structured incident management approach consists of identifying, logging, prioritizing, and closing incidents with clear role assignments.
Postmortems	Conducting postmortems encourages a blame-free culture and fosters genuine improvements through incident understanding.
Wrap-Up	To improve alerts, on-call practices, and incident management: avoid email alerts, maintain runbooks, evaluate alert criteria, and implement automation.

Chapter 3: Alerts, On-Call, and Incident Management

Importance of Alerts

Alerts are essential for monitoring systems, especially when infrastructure issues occur unexpectedly, often at inconvenient times. The primary purpose of monitoring extends beyond merely sending alerts; it involves observing and understanding system behavior over time.

Challenges of Effective Alerting

Great alerting is complex due to:
- Variability in system metrics leading to false alarms.
- The need for alerts to reach humans who have limited attention spans.

Characteristics of a Good Alert

A good alert should:

Why Bookey is must have App for Book Lovers

30min Content
The deeper and clearer interpretation we provide, the better grasp of each title you have.

Text and Audio format
Absorb knowledge even in fragmented time.

Quiz
Check whether you have mastered what you just learned.

And more
Multiple Voices & fonts, Mind Map, Quotes, IdeaClips...

Chapter 4 Summary : 4. Statistics Primer

Chapter 4: Statistics Primer

Introduction

Statistics is often undervalued and misunderstood in software engineering and systems administration. Many believe that simply applying statistical methods will yield magical insights. However, a fundamental understanding of statistics is straightforward and valuable for monitoring.

Before Statistics in Systems Operations

The effectiveness of monitoring tools like Nagios is limited because alerts typically compare current values against preset thresholds without considering data trends. This approach can lead to excessive noise in alerts. The chapter advocates recording data over time in time series databases, allowing for better statistical analysis and problem detection.

Math to the Rescue!

Modern monitoring systems emphasize the importance of preserving metrics for statistical analysis. Collecting data at regular intervals enables more informed checks against historical values rather than just current thresholds. This leads to improved insights through arithmetic and statistical functions.

Statistics Isn’t Magic

Successful use of statistics requires more than just simple application; it involves careful consideration of the right approach for the data. The chapter covers essential statistical principles, aiming to clarify common misconceptions and provide foundational knowledge.

Mean and Average

The mean (average) provides a general understanding of a dataset. Moving averages smooth out data and help in visualizing trends over time. However, excessive smoothing can hide valuable data points.

Median

The median represents the middle value in a dataset and is useful for skewed data, where the mean may not accurately reflect the central tendency. The median remains stable even in the presence of outliers.

Seasonality

Seasonality describes repeating patterns in data, such as daily commute times. Understanding seasonality aids in planning and predictions, allowing for more accurate monitoring insights over time.

Quantiles

Quantiles, particularly percentiles, represent specific points within a dataset. Percentiles are valuable in analyzing trends, such as bandwidth billing, by providing insights while ignoring outliers. However, calculating averages from percentiles is invalid because data points are omitted during this process.

Standard Deviation

Standard deviation provides insight into how values distribute around the mean, but it is reliable only in normally distributed datasets. Many real-world datasets are not normally distributed, making this metric less useful.

Wrap-Up

This chapter introduces key statistical concepts vital for monitoring. While averaging and median values are commonly applicable, seasonality and percentiles help grasp data patterns. Caution should be exercised when using statistical methods, particularly standard deviation, as many datasets may not fit the expected distribution patterns. Moving forward, consider your data's characteristics to select the most appropriate statistical approaches.

Conclusion

The chapter sets the foundation for applying statistics effectively in monitoring. It prepares readers for the next sections, which delve into practical monitoring strategies.

Critical Thinking

Key Point:Statistics as a Tool for Monitoring

Critical Interpretation:The chapter emphasizes that statistics are vital for effective monitoring, yet the author's assertion overlooks the complexity of data context and the variability in statistical interpretation. While advocacy for statistical methods can enhance monitoring efficiency, it may not universally lead to better outcomes without considering the unique nuances of each dataset or operational environment. Critics like Edward Tufte in "The Visual Display of Quantitative Information" argue for a holistic understanding of data visualization that often surpasses basic statistical measures, suggesting that simplistic adoption of such statistics might obscure rather than clarify important insights.

Chapter 5 Summary : 5. Monitoring the Business

Section	Summary
Introduction to Business Monitoring	Focuses on the need to align monitoring efforts with business KPIs to better address executive concerns compared to those of engineers.
Understanding Business KPIs	Discusses the importance of KPIs in measuring a company's health, covering metrics like MRR, NPS, and Customer Churn Rate that executives care about.
Real-World Examples	Yelp and Reddit are examples of companies using metrics to gauge app health and user engagement, demonstrating the importance of business KPIs as leading indicators.
Tying Business KPIs to Technical Metrics	Stresses the importance of connecting business KPIs to technical metrics, such as login success rates, to gain insights into app functionality.
Finding Your Company’s Business KPIs	Encourages collaboration with product managers and engineering teams to identify company-specific KPIs, emphasizing there are no universal metrics.
Wrap-Up	Reiterates the importance of tracking business KPIs for app performance and overall business health, setting the stage for future discussions on frontend performance monitoring.

Chapter 5: Monitoring the Business

Introduction to Business Monitoring

In Chapter 5, the focus shifts to monitoring from a business perspective, emphasizing the need to understand the questions that executives ask compared to those of engineers. By aligning monitoring efforts with business KPIs, engineers can more effectively address high-leverage problems.

Understanding Business KPIs

Key Performance Indicators (KPIs) measure a company's performance in areas deemed vital for overall health. Common questions from executives revolve around customer usability, profitability, growth trends, and customer satisfaction. Metrics that capture these inquiries include:

- Monthly Recurring Revenue (MRR)
- Revenue per Customer
- Number of Paying Customers
- Net Promoter Score (NPS)
- Customer Lifetime Value (LTV)
- Cost per Customer
- Customer Acquisition Cost (CAC)
- Customer Churn Rate
- Active Users
- Burn Rate
- Run Rate
- Total Addressable Market (TAM)
- Gross Profit Margin

These metrics provide a way to gauge business health, often requiring judgment.

Real-World Examples

To illustrate the concepts, the chapter presents two companies:

1.
Yelp
: Tracks metrics such as searches performed, reviews placed, active businesses, and ads purchased. These indicators help gauge the app's health and user engagement.

2.
Reddit
: Measures metrics like active users, user logins, threads submitted, and ads purchased, which highlight engagement and functionality.

Both examples demonstrate how business KPIs serve as leading indicators of overall performance, which can flag potential backend issues.

Tying Business KPIs to Technical Metrics

The chapter emphasizes the importance of connecting business KPIs to technical metrics. For example, tracking login success and failure rates can provide insight into user

login performance, which is crucial for understanding app functionality.

Finding Your Company’s Business KPIs

To identify relevant KPIs, it’s recommended to engage with product managers and engineering teams. Key questions to ask include:
- How do we know the app is working?
- What are the essential KPIs?
Mapping the app's functionality helps solidify understanding of what to measure. The chapter reinforces that while there are no universal metrics, identifying company-specific KPIs is essential for effective monitoring.

Wrap-Up

Chapter 5 reinforces the significance of recognizing and tracking business KPIs as they are critical for understanding app performance and ensuring the health of the business. Future chapters will delve into frontend performance monitoring.

Example

Key Point:Connecting business goals with technical metrics is vital for effective monitoring.

Example:Imagine navigating your application’s performance using only technical metrics like server uptime. While these numbers are relevant, without aligning them with business KPIs such as customer satisfaction or revenue per user, it's akin to steering a ship without knowing your destination. You might be afloat, but if customer churn is high or acquisition costs are soaring, you’re bound to drift off course. Integrating these insights enables you to understand not just that the system works, but how well it serves your business's growth and sustainability, ensuring you’re steering in the right direction.

Critical Thinking

Key Point:The alignment of technical and business KPIs is crucial for effective monitoring.

Critical Interpretation:While the author emphasizes the need for aligning monitoring efforts with business KPIs, one could argue that this perspective may overlook the nuance of specific sectors or unique business models that do not fit neatly into generalized metrics. The assumption that a unified set of KPIs applies universally to all businesses could lead to oversights in specialized industries where conventional business measures don't capture essential performance indicators. Alternative frameworks or methodologies, such as balanced scorecards or industry-specific metrics, as suggested by various business management sources, could provide a more comprehensive view and should be considered to avoid a one-size-fits-all approach.

Chapter 6 Summary : 6. Frontend Monitoring

Chapter 6: Frontend Monitoring

Introduction

Many companies neglect frontend monitoring, often viewing it as solely the responsibility of operations. This oversight leads to significant blind spots that can negatively impact application performance. This chapter discusses the importance of frontend monitoring and explores various approaches to effectively integrate it into existing monitoring strategies.

What is Frontend Monitoring?

Frontend monitoring encompasses elements executed on the client-side, such as HTML, CSS, JavaScript, and images. Traditional monitoring methods struggle to address the increasing workload managed on the frontend, especially

with the rise of single-page applications (SPAs) that can experience client-side errors without corresponding server issues.

Understanding Single-Page Applications (SPAs)

SPAs are web applications where most resources load client-side, minimizing server requests. This design allows for seamless data updates without unnecessary page refreshes. Monitoring practices need to adapt to the unique performance characteristics of SPAs.

Importance of Frontend Performance

Slow applications can have a detrimental impact on business outcomes. Studies show that even a one-second delay in load time can lead to substantial losses in page views, conversions, and customer satisfaction. Businesses should

Positive feedback

Sara Scholz

tes after each book summary
erstanding but also make the
and engaging. Bookey has
ding for me.

Fantastic!!!

Masood El Toure

I'm amazed by the variety of books and languages Bookey supports. It's not just an app, it's a gateway to global knowledge. Plus, earning points for charity is a big plus!

Fi
Ab
bo
to
m

José Botín

ding habit
p's design
ual growth

Love it!

Wonnie Tappkx

Bookey offers me time to go through the important parts of a book. It also gives me enough idea whether or not I should purchase the whole book version or not! It is easy to use!

Time saver!

Bookey is my go-to app for
summaries are concise, ins
curated. It's like having acc
right at my fingertips!

Awesome app!

Rahul Malviya

I love audiobooks but don't always have time to listen to the entire book! bookey allows me to get a summary of the highlights of the book I'm interested in!!! What a great concept !!!highly recommended!

Beautiful App

Alex Walk

This app is a lifesaver for book lovers with busy schedules. The summaries are spot on, and the mind maps help reinforce wh I've learned. Highly recommend!

Chapter 7 Summary : 7. Application Monitoring

Chapter 7: Application Monitoring

Overview

Many organizations excel in monitoring their server infrastructure, security, and network, yet overlook application monitoring, treating it as an enigma. Given the frequent updates applications undergo, visibility into their performance is crucial. The misconception exists that application monitoring is complex and requires specialized skills, a notion this chapter aims to dispel.

Instrumenting Your Apps with Metrics

Instrumenting applications for metrics is vital for proactive performance maintenance. Initial steps should involve tracking simple metrics, such as database query times and user login counts. This practice quickly becomes valuable

and often addictive.

Application Performance Monitoring (APM) Tools

APM tools can automatically gather data about application performance, but they lack context about the specific application behavior and business logic. Although these tools have their merits, understanding their limitations is essential. StatsD is highlighted as a simple and effective tool for adding metrics to code, helping to illustrate the ease of instrumenting applications.

Monitoring Build and Release Pipelines

Monitoring the build and release process provides significant insights and helps identify regressions. Tracking deploy events (start, end, and responsible person) alongside application metrics offers a holistic understanding of performance changes related to deployments.

Health Endpoint Pattern

This concept refers to an HTTP endpoint in an application that reports its health and state. Such endpoints can be

utilized for load balancing, debugging, and monitoring dependencies. Although this requires more engineering effort than simple metrics, its benefits, such as enhanced debugging capabilities, are substantial.

Application Logging

Logging is a crucial counterpart to metrics, allowing for a deeper understanding of application behavior. Structured logs, particularly in JSON, are recommended for clarity and ease of analysis. The debate between metrics and logs hinges on which approach fits the monitoring context best, emphasizing the need for thoughtful logging and metric selection.

Security and Implementation Challenges

While health endpoints offer various benefits, security concerns arise regarding public access. Furthermore, implementing complex checks can complicate debugging processes. The endpoint's implementation should use proper HTTP codes to facilitate easy status assessment.

Serverless and Microservice Monitoring

Monitoring serverless functions and microservices presents unique challenges due to their ephemeral nature and intricate inter-service communication. Distributed tracing is recognized as an essential but challenging technique for monitoring interactions in microservice architectures, allowing for tracing request paths across services.

Wrap-Up

Application monitoring, through metrics and logs, significantly improves troubleshooting capabilities. Tracking releases and performance correlations enhances operational oversight. Although the health endpoint pattern and distributed tracing are valuable, the foundational monitoring practices remain crucial for understanding application performance.

Chapter 8 Summary : 8. Server Monitoring

Chapter 8: Server Monitoring

Introduction to Server Monitoring

- Monitoring is often associated with sysadmins but encompasses much more, including various metrics and logs from servers.
- This chapter focuses on common services, metrics, and logs found in modern server architectures, particularly using Linux as the basis for discussion.

Standard OS Metrics

- The standard OS metrics (CPU, memory, load, network, disk) are often misunderstood and should not be the primary focus for alerts.
- These metrics can be useful for diagnostics and troubleshooting if used in the right context.

- Recommendations include recording these metrics but avoiding alerts on them unless necessary.

Key Metrics for Monitoring

CPU Monitoring

- CPU utilization can be found in `/proc/stat` and tools like `top`.
- The percentage of active CPU time is calculated from various states of CPU operations.

Memory Monitoring

- Key aspects include memory used vs. free, with distinctions among shared, cached, and buffered memory.
- The second row of memory metrics is vital for determining actual usage excluding transient buffers/cache.

Network Monitoring

- Network metrics derive from `/proc/net/dev` using tools like `ifconfig` and `ip`.

- Important metrics to track include in/out octets, errors, and drops.

Disk Performance Monitoring

- Disk performance is analyzed through `/proc/diskstats` using `iostat`.
- Key metrics include `await` time and `%util`, indicating the wait time and disk utilization.

Load Monitoring

- Load refers to processes waiting for CPU resources, shown as averages over different periods.
- Importantly, load metrics don't directly translate to performance issues and should be interpreted cautiously.

SSL Certificate Monitoring

- Monitoring SSL certificates is crucial to avoid expiration issues.
- Alerts should be set up for certificate expirations through monitoring tools or scripts.

Avoiding SNMP

- SNMP (Simple Network Management Protocol) is discouraged due to its complexity and inherent insecurities.
- Alternatives such as push-based tools (e.g., collectd, Telegraf) are recommended.

Web Server Performance Monitoring

- Key metrics include requests per second (req/sec) and HTTP status codes.
- Monitoring for non-200 responses and connection metrics is essential for understanding web server health.

Database Server Monitoring

- Focus on the number of connections (MySQL calls this "threads") and queries per second (qps).
- Monitoring slow queries and IOPS (input/output operations per second) is critical for database performance.

Load Balancer Monitoring

- Similar metrics to web servers are monitored, focusing on

frontend and backend health.
- Health checks are essential for ensuring the backend servers are functioning properly.

Message Queue Monitoring

- Monitor queue length and consumption rate to gauge the message flow in a queue.
- Abnormal lengths or consumption rates can indicate underlying issues in message processing.

Caching Metrics

- Metrics include the number of evicted items and hit/miss ratios.
- Tracking these provides insight into cache performance and efficiency.

DNS and NTP Monitoring

- DNS performance is essential if managing your own servers, focusing on zone transfers and query loads.
- NTP (Network Time Protocol) sync status is crucial for time-dependent operations.

Corporate Infrastructure Monitoring

- DHCP and email services are vital for traditional setups.
- Monitor lease capacity and outgoing email queues for potential issues.

Monitoring Scheduled Jobs

- Use data presence to monitor cron jobs and scheduled tasks to ensure they run successfully.

Logging

- Effective logging is divided into collection, storage, and analysis.
- Use centralized log management solutions for better analysis rather than relying on syslog alone.

Wrap-Up

- This chapter covers various aspects of server monitoring, emphasizing the importance of context and proper interpretation of metrics both for diagnostics and operational efficiency.

Chapter 9 Summary : 9. Network Monitoring

Chapter 9: Network Monitoring

Introduction

Network monitoring is essential in IT, as demonstrated by the author's early experiences. Understanding network behavior is critical for the performance of all dependent applications.

The Pains of SNMP

SNMP (Simple Network Management Protocol) is fundamental for monitoring network performance but feels outdated. Despite its challenges, it remains the primary method for managing network devices.

What is SNMP?

Introduced in 1988, SNMP is a protocol for monitoring

devices across a network. While system administration has evolved, SNMP has been largely static.

How Does It Work?

SNMP relies on a manager-agent model, where the manager queries the agent (the device being monitored) for information using object identifiers (OIDs).

Monitoring SNMP

Monitoring mainly relies on polling agents and using traps to capture events. Errors in agent implementations and varying SNMP versions (1, 2, and 3) affect usability and security.

Security Considerations

SNMP is inherently insecure, but SNMP version 3 offers

Read, Share, Empower

Finish Your Reading Challenge, Donate Books to African Children.

The Concept

BOOKS FOR AFRICA × ×

This book donation activity is rolling out together with Books For Africa. We release this project because we share the same belief as BFA: For many children in Africa, the gift of books truly is a gift of hope.

The Rule

Earn 100 points → **Redeem a book** → **Donate to Africa**

Your learning not only brings knowledge but also allows you to earn points for charitable causes! For every 100 points you earn, a book will be donated to Africa.

Chapter 10 Summary : 10. Security Monitoring

Chapter 10: Security Monitoring

Overview of Security Monitoring

Security monitoring differs significantly from traditional infrastructure or application monitoring. Unlike existing system metrics, security lacks inherent hooks or instrumentation, often compelling engineers to retrofit security measures into pre-existing systems. This chapter explores the basics of building security frameworks while stressing that security monitoring is a specialized field deserving deeper study.

The Continuum of Security

Security is viewed as a continuum, ranging from minimal (like a wet paper bag) to maximum protection (like Fort Knox). Proper security implementation should balance costs,

risks, and workflow interruptions. Understanding the nature of threats and the level of acceptable security is crucial.

Monitoring and Compliance

Compliance with regulations (e.g., HIPAA, PCI-DSS, SOC2) necessitates monitoring various security controls. This includes ensuring specific requirements are met through demonstrable monitoring, such as tracking connections and maintaining antivirus mechanisms. The monitoring obligation encompasses vast scopes, leading to a recommendation for comprehensive logging.

User, Command, and Filesystem Auditing with auditd

auditd is a userspace interface for the Linux Audit System designed to track security-focused events. It logs user actions and can be configured to monitor specific events like sudo executions and file accesses. The configuration involves setting custom rules to define what to log.

Remote Log Management

To enhance security, logs should be forwarded to a centralized server using the audisp-remote plugin, ensuring logs are preserved against tampering. This method leverages auditd's independent operation compared to traditional syslog services.

Host Intrusion Detection System (HIDS)

HIDS monitors for malicious activities or breaches within a host, with a specific focus on rootkit detection. Tools like rkhunter employ various detection methods, including file hash comparisons and signature detection. Configuring rkhunter is straightforward and should be automated for regular monitoring.

Network Intrusion Detection System (NIDS)

NIDS detects and reports on threats within the network. Strategically placed network taps help intercept traffic for analysis. The recommendation is to use hardware taps over span ports due to their higher resilience in handling traffic loads. Regular tuning of NIDS is required to maximize its effectiveness.

Conclusion

The chapter encapsulates foundational aspects of security monitoring, emphasizing regulatory compliance, user activity auditing, rootkit detection, and network threat identification. Further exploration into these areas is encouraged for effective security management.

Chapter 11 Summary : 11. Conducting a Monitoring Assessment

Chapter 11: Conducting a Monitoring Assessment

This chapter serves as a comprehensive guide to performing a monitoring assessment, drawing on lessons from previous chapters. It emphasizes the importance of systematically determining what to monitor in order to gain a clearer understanding of an application's behavior and its underlying infrastructure.

Business KPIs

To begin, Tater.ly's mission, which is to help users find the best french fries via restaurant reviews, establishes the context for identifying key performance indicators (KPIs). Key metrics include:
- Number of restaurants reviewed
- Number of active restaurants
- Number of users and active users
- Searches performed

- Reviews placed
- Ads purchased
- Direction and rate of change for all metrics
- Net promoter scores (NPS) from both users and restaurants

Frontend Monitoring

The focus here is primarily on Real User Monitoring (RUM) metrics to keep track of page load times from the user perspective, indicating the health of the user experience.

Application and Server Monitoring

An architecture diagram of Tater.ly's infrastructure provides insight into its standard three-tier architecture. Key metrics and logs to monitor include:
- Page load time
- User logins (successes and failures)
- Searches performed
- Reviews submitted
- PostgreSQLquery latency
- Redis performance metrics
- CDN performance

Logs should include success/failure reasons for user logins

and server-side daemon logs (e.g., Apache, PostgreSQL).

Security Monitoring

Security monitoring focuses on straightforward aspects like SSH login attempts and syslog/audit logs since Tater.ly is not bound by compliance requirements.

Alerting

Effective alerts are crucial; based on the identified metrics and logs, essential alerts should cover:
- Increases in page load time
- Increases in error rates and latency across various components
- Increases in latency for specific application actions (searches, reviews, logins)
Creating runbooks for the identified metrics would help share knowledge amongst colleagues.

Wrap-Up

This concludes the first monitoring assessment. Although just a starting point, it highlights that monitoring is an

ongoing process as business needs and infrastructure evolve over time. Continuous improvement is key.

Chapter 12 Summary : A. An Example Runbook: Demo App

Section	Details
Appendix Title	An Example Runbook: Demo App
Overview	The Rails Demo App is a simple blogging application showcasing user management and a post/comment system.
Metadata	Codebase Location: demo-app Service Owner: John Doe
Escalation Procedure	First escalation point: John Doe (refer to the company contact sheet for details).
Dependencies	External: None Internal: PostgreSQL database on RDS (rds-123.foo.com)
Tech Stack	Rails 4.x PostgreSQL (AWS RDS)
Metrics and Logs	Metrics: User login/logout counts Post/comment creation/deletion counts Timers for post creation, deletion, signup, and login/logout Logs: User sign-in attempts with ID, status, and IP Post and comment creation attempts with similar details
Alerts	User Sign-in Failure Rate: Exceeds 5% in 5 minutes. User Login Time Too High: Exceeds one second. Post Create Time Too High: Exceeds one second.

Section	Details
	Comment Create Time Too High: Exceeds one second.
Purpose	This runbook example offers a structured approach to monitoring and incident response, highlighting metrics, logs, and alerts.

Appendix A: An Example Runbook: Demo App

This chapter presents a sample runbook designed for use in your environment, serving as a foundational template to enhance over time. A well-structured runbook is vital for effective monitoring and incident response.

Demo App Overview

The Rails Demo App is a straightforward blogging application built with Rails, demonstrating standard components including user management and a post/comment system.

Metadata

-

Codebase Location:

Found in the internal source code system under the name

`demo-app`.

-

Service Owner:
John Doe.

Escalation Procedure

For issue resolution, the service owner, John Doe, should be the first escalation point. Refer to the company contact sheet for further instructions.

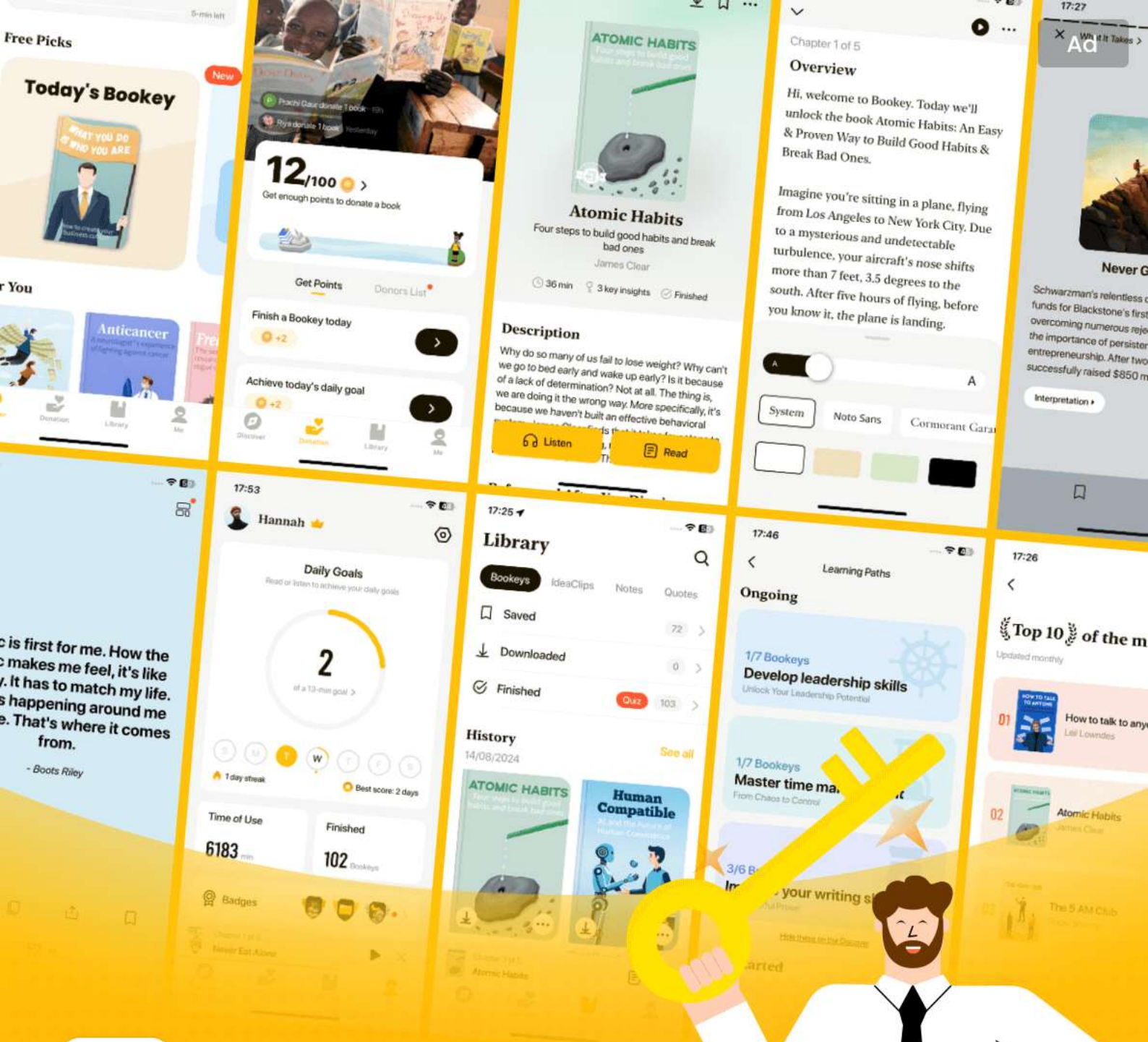
Dependencies

-

External Dependencies:
None

-

Internal Dependencies:



World' best ideas unlock your potential

Free Trial with Bookey

Download on the App Store

GET IT ON Google Play

Scan to download

Chapter 13 Summary : B. Availability Chart

Appendix B: Availability Chart

Overview

This appendix presents a chart of availability numbers, providing a reference for assessing allowable downtime within specific availability targets.

Availability Percentages and Corresponding Downtime

-

90% ("one nine")

- Downtime per year: 36.5 days
- Downtime per month: 72 hours
- Downtime per week: 16.8 hours
- Downtime per day: 2.4 hours

-
95% ("one and a half nines")

  - Downtime per year: 18.25 days
  - Downtime per month: 36 hours
  - Downtime per week: 8.4 hours
  - Downtime per day: 1.2 hours
-
97%

  - Downtime per year: 10.96 days
  - Downtime per month: 21.6 hours
  - Downtime per week: 5.04 hours
  - Downtime per day: 43.2 minutes
-
98%

  - Downtime per year: 7.30 days
  - Downtime per month: 14.4 hours
  - Downtime per week: 3.36 hours
  - Downtime per day: 28.8 minutes
-
99% ("two nines")

  - Downtime per year: 3.65 days
  - Downtime per month: 7.20 hours
  - Downtime per week: 1.68 hours
  - Downtime per day: 14.4 minutes
-
99.5% ("two and a half nines")

  - Downtime per year: 1.83 days
  - Downtime per month: 3.60 hours
  - Downtime per week: 50.4 minutes
  - Downtime per day: 7.2 minutes
-
99.8%

  - Downtime per year: 17.52 hours
  - Downtime per month: 86.23 minutes
  - Downtime per week: 20.16 minutes
  - Downtime per day: 2.88 minutes
-
99.9% ("three nines")

  - Downtime per year: 8.76 hours
  - Downtime per month: 43.8 minutes
  - Downtime per week: 10.1 minutes

  - Downtime per day: 1.44 minutes
-
99.95% ("three and a half nines")

  - Downtime per year: 4.38 hours
  - Downtime per month: 21.56 minutes
  - Downtime per week: 5.04 minutes
  - Downtime per day: 43.2 seconds
-
99.99% ("four nines")

  - Downtime per year: 52.56 minutes
  - Downtime per month: 4.38 minutes
  - Downtime per week: 1.01 minutes
  - Downtime per day: 8.64 seconds
-
99.995% ("four and a half nines")

  - Downtime per year: 26.28 minutes
  - Downtime per month: 2.16 minutes
  - Downtime per week: 30.24 seconds
  - Downtime per day: 4.32 seconds
-
99.999% ("five nines")

  - Downtime per year: 5.26 minutes
  - Downtime per month: 25.9 seconds
  - Downtime per week: 6.05 seconds
  - Downtime per day: 864.3 milliseconds
-
99.9999% ("six nines")

  - Downtime per year: 31.5 seconds
  - Downtime per month: 2.59 seconds
  - Downtime per week: 604.8 milliseconds
  - Downtime per day: 86.4 milliseconds
-
99.99999% ("seven nines")

  - Downtime per year: 3.15 seconds
  - Downtime per month: 262.97 milliseconds
  - Downtime per week: 60.48 milliseconds
  - Downtime per day: 8.64 milliseconds
-
99.999999% ("eight nines")

  - Downtime per year: 315.569 milliseconds
  - Downtime per month: 26.297 milliseconds

- Downtime per week: 6.048 milliseconds
- Downtime per day: 0.864 milliseconds

-

99.9999999% ("nine nines")

- Downtime per year: 31.5569 milliseconds
- Downtime per month: 2.6297 milliseconds
- Downtime per week: 0.6048 milliseconds
- Downtime per day: 0.0864 milliseconds

Best Quotes from Practical Monitoring by Mike Julian with Page Numbers

Chapter 1 | Quotes From Pages 29-71

1. Too many security organizations put tools before operations. They think, 'we need to buy a log management system', or 'I will assign one analyst to antivirus duty, one to data leakage protection duty.' A tool-driven team will not be effective as a mission-driven team.
2. If you learn nothing else from this book, remember this: there are no silver bullets. Anything worth solving takes a bit of effort, and monitoring a complex system is certainly no exception.
3. Monitoring isn’t just a single, cut-and-dry problem—it’s actually a huge problem set.
4. Creating your own specialized tool does have its advantages. For example, one of the first tools many teams build is something to allow the creation of AWS EC2

instances quickly and with all the standards of their company automatically applied.

5. You wouldn’t expect only one member of your team to be the sole person familiar with your config management tool, or how to manage your database servers, so why would you expect that when it comes to monitoring?
6. More monitoring doesn’t fix a broken system, and it’s not an improvement in your situation.
7. If you cannot quickly configure new checks or nodes, building better monitoring becomes frustrating. After a while, you’ll just stop bothering.
8. Don’t adopt them simply because a well-known company uses them. It is important to evaluate and prototype solutions rather than choosing them because someone else uses them or because a team member used them in the past.

Chapter 2 | Quotes From Pages 72-174

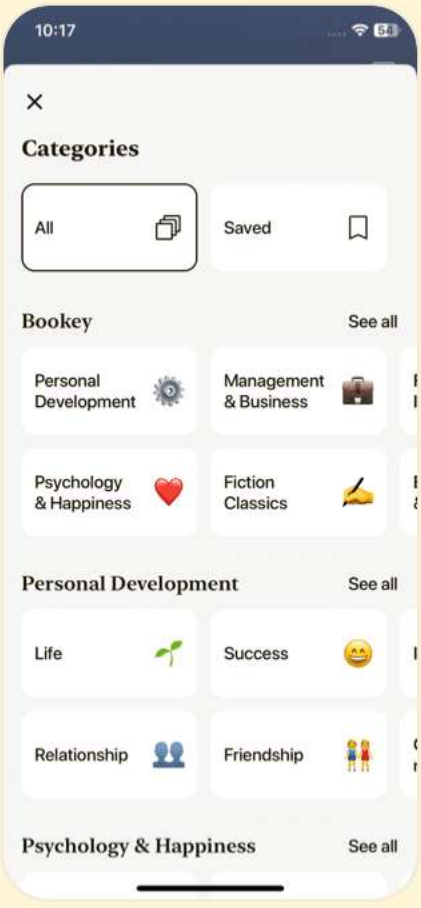
1. Monitoring is for asking questions." - Dave Josephsen, Monitorama 2016
2. Start monitoring as close to the user as possible.

3. You should be building things in a way that works best for your environment.
4. Always be improving.
5. It’s better to use SaaS solutions for monitoring than to build your own tools.
6. You’re (Probably) Not an Expert at Architecting These Tools.
7. World-class isn’t achieved in a week, but rather, over months and years of consistent attention and improvement.

Chapter 3 | Quotes From Pages 175-220

1. Without alerts, we’d all have to be staring at graphs all day long, every day.
2. Great alerting is harder than it seems.
3. An alert should evoke a sense of urgency and require action from the person receiving that alert.
4. The solution to alert fatigue is simple on its face: fewer alerts.
5. A runbook is for when human judgment and diagnosis is necessary to resolve something.

6. In fact, I’d bet that a lot of you are doing something very similar to this already, and that’s great.
7. You need to fix things after they break.
8. You’ve got to fix your shit.
9. Establishing your incident response as an internal standard exposes patterns and hot spots in the app and infrastructure.
10. Don’t send alerts to email.



Chapter 4 | Quotes From Pages 221-248

1. Statistics isn’t magic.
2. By asking these questions of your data, you’ll start to understand which statistical approaches may work well and which may not.
3. Mean, more commonly known as average, is useful for determining what a dataset generally looks like without examining every single entry in the set.
4. Determining the correct amount of smoothing to apply is a balancing act.
5. Percentiles are helpful for understanding what the bulk of your data looks like, but be careful: they inherently ignore the extreme datapoints.

Chapter 5 | Quotes From Pages 251-277

1. Monitoring from the user’s perspective... Is the site up? Are users impacted?
2. Once we learn to ask the questions the executives are asking, we can really begin to work on the most important and highest-leverage problems facing the business.

3.Business KPIs are among the most important metrics out there and make great leading indicators for the health and performance of your app and infrastructure.

4.To get visibility into the performance of your app and infrastructure, you have to have a design for it.

5.I have a foolproof way of ensuring you understand how the app works and what's important to measure: talk to people.

Chapter 6 | Quotes From Pages 278-312

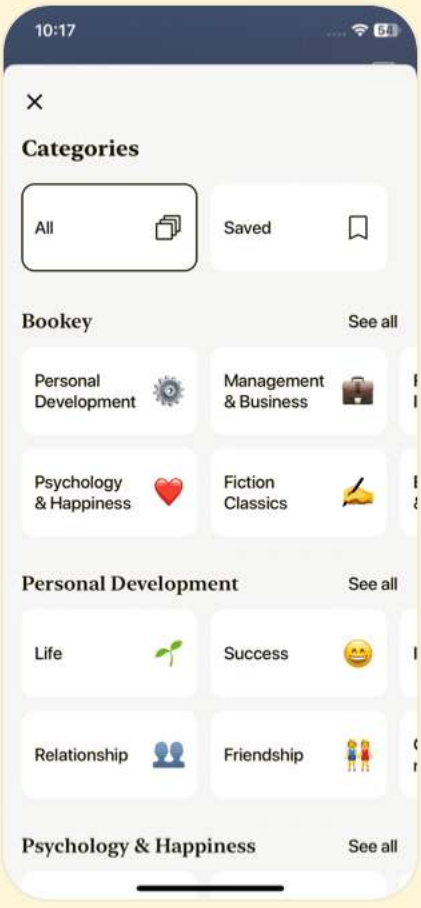
1. As engineers, we intuitively understand that a slow app is bad for business.

2.Don't be one of those teams: great site performance is a requirement for profitable businesses that sell something online.

3.Because the traffic and conversion rate increases are multiplicative, this was a huge win for us in terms of web and app signups.

4.Performance of web apps tends to degrade over time, unless you actively optimize for performance on a regular basis.

5. Monitor page load times for actual users. Monitor for JavaScript exceptions. Keep track of page load time over time with your CI system, ensuring load times stay within an acceptable range.



Chapter 7 | Quotes From Pages 313-355

1....their applications are an unknowable black box.

2.One of the most powerful things you can do in monitoring is also one of the most overlooked things: instrumenting your own applications.

3.These tools have zero context about your app or the business logic behind it.

4.An important bit about the flush is that all metrics collected during the flush interval are aggregated and then sent to the backend.

5....the /health endpoint pattern...is helpful for debugging: exposing build information in the endpoint helps with determining what is running in the environment easily.

6.Metrics can only tell you so much about what your application is up to, which is why it’s important to also be logging behavior and actions from your applications.

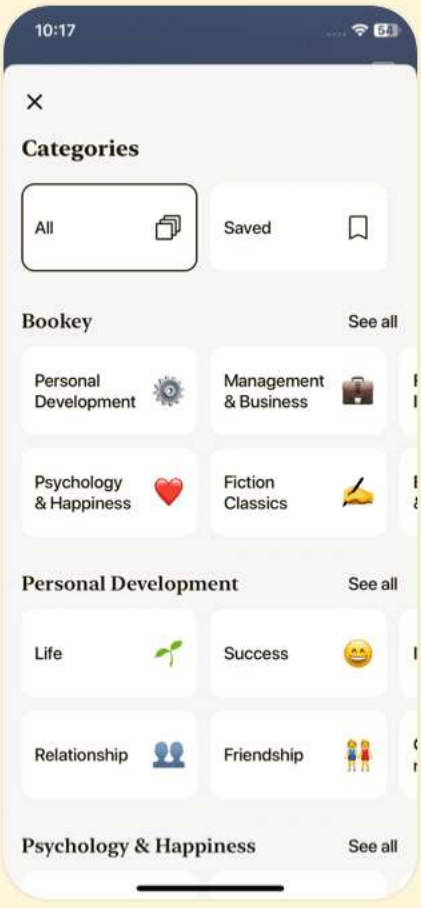
7.Spend the time to think through the app, and the log statements you need will become obvious.

Chapter 8 | Quotes From Pages 356-422

1. This is unfortunate since we've seen there's so much more to monitoring than just what happens on a server.
2. In order to know if things are working, you have to start at the top instead.
3. These metrics are some of the most powerful metrics you have available.
4. High iowait is something we want to avoid.
5. The one exception to this is that load makes for a somewhat decent proxy metric.
6. Nothing is more frustrating than troubleshooting slow database performance and finding out an hour into it that it's just a failing disk that's caused IOPS to drop.
7. Most tools report memory metrics based on values reported by /proc/meminfo.
8. I highly recommend reading Baron Schwartz's High Performance MySQL and Laine Campbell and Charity Majors' Database Reliability Engineering.

Chapter 9 | Quotes From Pages 423-511

1. The behavior and performance of the network is fundamental to the behavior and performance of everything that relies on it, which, these days, is everything.
2. Increasing the availability of the network is a nice lever you pull to allow everything that relies on it to improve.
3. Networking is one of the few 'dark arts' left in the tech world. So many people don’t understand it, yet it’s a crucial component of everything we do.
4. It’s been my experience that any data in an SNMP trap can also be found in the device's syslog, so I tend to disable traps entirely.
5. Tracking configuration changes yields lots of great information and can save you time and headaches.
6. Voice and video performance is all about three measurements: latency, jitter, and packet loss.
7. Monitoring the network is far more complex and involved than many realize, especially at large scale.



Chapter 10 | Quotes From Pages 512-537

1. Security is a matter of assessing threat and risk and deciding on compromises.
2. It’s important that you consciously make this decision instead of absently deciding that security is too difficult.
3. Monitoring requirements for compliance purposes are often more straightforward than they appear, though they may not always be easy.
4. Firewalls aren’t enough for network security.
5. Careful placement of network taps and NIDS will yield a wealth of information.

Chapter 11 | Quotes From Pages 538-553

1. Performing a monitoring assessment on your environment is a great way to systematically determine what you should be monitoring and why.
2. It’s by no means exhaustive or perfect, but rather, it’s intended to be a starting point to get you thinking about what matters and what doesn’t.

3.Looking at the metrics and logs we've identified, I would expect these alerts to be in place:

4.Congratulations, you just finished your first monitoring assessment— that wasn't so hard, was it?

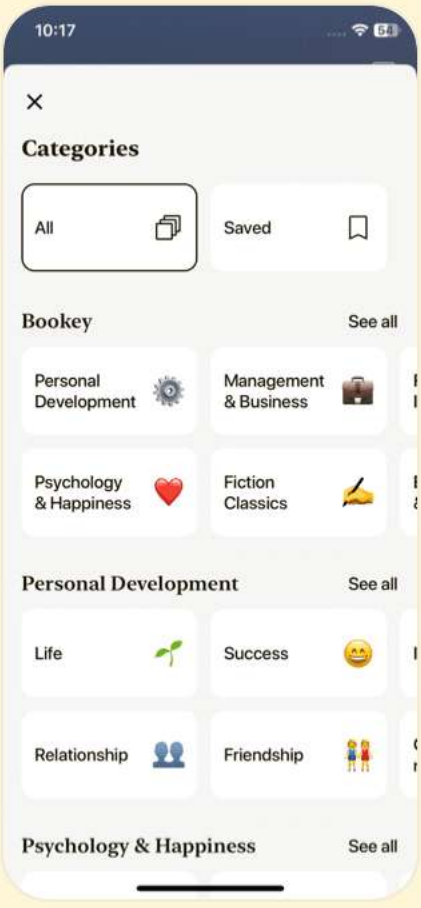
5.This assessment is a great starting point, but don't forget to keep improving.

Chapter 12 | Quotes From Pages 554-571

1. A runbook is only as good as the information in it, so if you find you need different sections, by all means, create them!

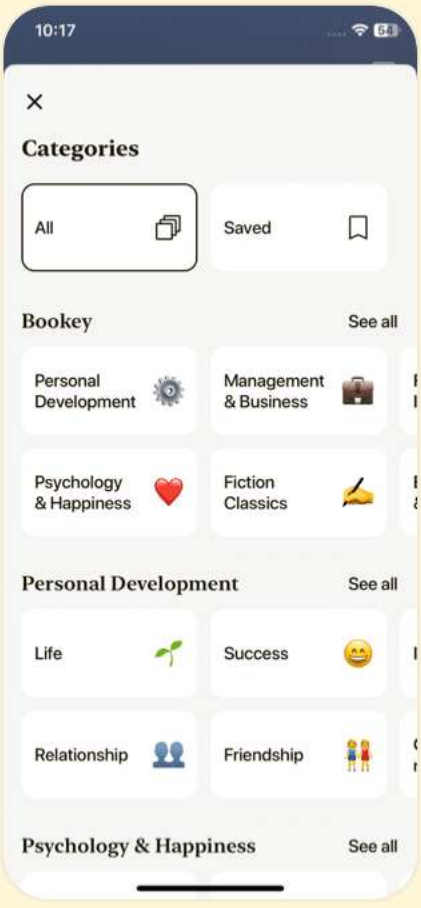
2.This is a great starting point, and I encourage you to build on this and iterate over time.

3.In the event assistance is needed to resolve an issue with this service, the service owner has requested to be the next escalation point.



Chapter 13 | Quotes From Pages 572-574

1. It's a great reference for how much downtime is allowed within a given availability target.
2. Availability % signifies how much time a system is operational versus downtime.
3. Each increase in availability percentage represents a significant decrease in allowable downtime.



Practical Monitoring Questions

View on Bookey Website

Chapter 1 | 1. Monitoring Anti-Patterns| Q&A

1.Question

What is the significance of identifying monitoring anti-patterns before implementing a monitoring solution?

Answer:Identifying monitoring anti-patterns is crucial because they represent bad habits and outdated practices that can undermine the effectiveness of your monitoring strategy. By recognizing and correcting these anti-patterns, you set a solid foundation for building a robust and effective monitoring system. This ensures that the monitoring efforts you implement will be aligned with modern best practices and capable of addressing the complexities of today's infrastructure.

2.Question

How can 'tool obsession' negatively impact a monitoring

strategy?

Answer:'Tool obsession' can lead organizations to focus excessively on acquiring new monitoring tools rather than developing operational capabilities and understanding the underlying needs of their systems. This results in a cycle of dissatisfaction, where teams constantly blame tools for monitoring deficiencies and keep seeking replacements instead of addressing root causes or improving practices. As a result, they fail to build a comprehensive monitoring strategy that integrates well with their operational goals.

3.Question

What does the 'observer effect' imply about monitoring systems, and why is it not a problem today?

Answer:The 'observer effect' suggests that observing a system can change its behavior. However, in modern environments, the additional load created by monitoring tools is minimal and manageable. Today's systems are designed to handle the minor overhead introduced by monitoring agents or tools, making it feasible and effective to observe and

collect data without significantly affecting performance.

4.Question

What does the chapter suggest about using multiple monitoring tools?

Answer:The chapter suggests that using multiple monitoring tools is not inherently negative; rather, it is essential to have a toolkit that includes both general and specialized tools to effectively monitor different aspects of a complex system. The key is to ensure that these tools can work together and that they are chosen thoughtfully, considering the specific monitoring needs.

5.Question

Why should monitoring not be relegated to a single person or role within a team?

Answer:Monitoring should not be limited to a single person or role because it requires a comprehensive understanding of the entire system, which is distributed among all team members. When everyone takes responsibility for monitoring, it fosters a culture of shared ownership and

ensures that monitoring strategies are informed by diverse perspectives. Additionally, this approach enhances the resilience of monitoring practices and contributes to more robust service performance.

6.Question

What is 'checkbox monitoring' and why should it be avoided?

Answer:'Checkbox monitoring' refers to the practice of implementing monitoring systems merely to tick boxes for compliance or assurance, without a genuine understanding of what needs to be monitored. This approach often leads to ineffective and noisy monitoring that provides little value. It should be avoided because it can create a false sense of security while ultimately failing to provide actionable insights or alerting capabilities when real issues arise.

7.Question

How can organizations ensure that their monitoring configurations are not overly manual?

Answer:Organizations should implement automation in their

monitoring configurations by utilizing tools like configuration management and self-service monitoring solutions. This reduces the burden of manual configuration and ensures that new nodes or services are automatically integrated into the monitoring framework, enabling faster responses to changes and minimizing human error.

8.Question

What is meant by 'using monitoring as a crutch,' and how can teams avoid this trap?

Answer:'Using monitoring as a crutch' means relying on monitoring tools to compensate for underlying problems within a system, rather than addressing the root causes of issues. Teams can avoid this by focusing on improving code quality, system stability, and resilience instead of merely adding more monitoring around failing systems. This proactive approach ensures that monitoring serves its intended purpose of providing insights into healthy operational performance.

9.Question

How should organizations determine the right number of monitoring tools to use?

Answer:Organizations should aim to use as few monitoring tools as necessary to get the job done effectively. They need to evaluate the purpose of each tool and eliminate redundancies, ensuring that each tool provides unique and valuable insights. Engaging in discussions with teams about existing tools and their functions can help identify overlaps and inform better tool management.

10.Question

What is the overall takeaway regarding monitoring anti-patterns as presented in Chapter 1?

Answer:The overall takeaway is that by identifying and addressing these monitoring anti-patterns—such as tool obsession, monitoring as a job, and checkbox monitoring—organizations can build healthier monitoring practices. This proactive adjustment leads to better visibility, performance, and ultimately, the ability to respond effectively to issues before they impact services.

Chapter 2 | 2. Monitoring Design Patterns| Q&A

1.Question

What is Composable Monitoring and why is it significant?

Answer:Composable Monitoring is a design pattern that advocates for using multiple specialized tools that work together loosely to form a flexible monitoring platform. This approach contrasts with monolithic tools like Nagios, which can be rigid and limit adaptability. By employing Composable Monitoring, teams can fully leverage specialized tools that best fit their needs, allowing for easier updates and replacements, leading to a more efficient and pain-free monitoring setup.

2.Question

How does the push model for data collection differ from the pull model?

Answer:In the push model, clients (like servers or applications) send data to a specified location actively, which

is beneficial for scaling in distributed architectures. Conversely, in the pull model, a central service requests data from clients at scheduled intervals, which can lead to challenges in managing and scaling due to the need for central scheduling. Push models are generally more flexible and require less coordination.

3.Question

Why is it essential to monitor from the user's perspective first?

Answer:Monitoring from the user's perspective ensures that you prioritize what truly impacts users, such as application performance and HTTP response codes. It helps you discern whether users are experiencing issues, irrespective of backend complexities. This approach helps teams focus on crucial metrics that directly affect user experience, thereby safeguarding service reliability.

4.Question

What are the core advantages of using SaaS for monitoring instead of building in-house solutions?

Answer:SaaS solutions for monitoring are generally cheaper and faster to implement than building an internal platform. They offer dedicated expertise without the overhead of hiring and training staff for custom solutions. In addition, SaaS tools often come with comprehensive user documentation and features that might take considerable time and effort to develop in-house.

5.Question

What is the importance of continual improvement in monitoring systems?

Answer:Monitoring systems are not static; they require ongoing evolution and refinement to adapt to changing needs of the organization and technological advancements. By constantly seeking improvements, organizations can better align their monitoring practices with actual business value while ensuring effectiveness over time.

6.Question

How can effective analytics and reporting contribute to better monitoring?

Answer:Effective analytics and reporting provide insights into service-level availability, enabling teams to understand performance trends over time. This understanding is crucial for meeting SLAs (Service Level Agreements) and making informed decisions to enhance service reliability and user satisfaction.

7.Question

What is a monitoring alert and what should one consider when creating it?

Answer:Alerts are warnings triggered when specific metrics exceed or fall below predefined thresholds. However, not every metric needs to have an alert associated with it. It's important to focus on alerts that truly reflect user impact rather than overwhelming teams with alerts that may not correlate with significant issues.

8.Question

Why is it crucial to think beyond just generating alerts in monitoring?

Answer:Focusing solely on generating alerts can lead to a

reactive approach rather than a proactive one. Monitoring should be about asking the right questions, providing insights into system performance, and facilitating improvements rather than merely responding to problems as they arise.

9.Question

What role does data storage play in a monitoring platform?

Answer:Data storage is vital for maintaining the metrics and logs collected over time. Effective storage solutions, such as Time Series Databases (TSDBs), enable quick access to historical data for analysis. Properly managing storage can enhance performance, prevent data loss, and facilitate better reporting, ultimately improving the monitoring system.

10.Question

How should logs be structured for optimal monitoring?

Answer:Logs should ideally be structured to facilitate easier parsing and analysis. Using formats like JSON allows for explicit mappings of field meanings, making it simpler for automated tools to extract valuable insights. While structured

logs are preferable for most use cases, it’s important to assess whether unstructured logs are more appropriate based on specific needs.

11.Question

What are some common pitfalls in managing SLAs?

Answer:Some pitfalls include setting unrealistic SLA expectations, misunderstanding the calculation of availability, and neglecting to report truthful metrics. Companies must recognize the balance between maintaining high availability and the practical realities of their infrastructure capabilities.

Chapter 3 | 3. Alerts, On-Call, and Incident Management| Q&A

1.Question

What is the primary purpose of alerts in a monitoring system?

Answer:The primary purpose of alerts in a monitoring system is to notify relevant personnel of issues that require immediate action to prevent system failures or to address outages.

2.Question

Why is alert fatigue a significant issue in incident management?

Answer:Alert fatigue becomes significant because it leads to desensitization towards alerts, causing personnel to ignore or overlook critical warnings due to overwhelming noise from frequent alerts.

3.Question

What distinguishes an 'alert' from an 'FYI' message?

Answer:An 'alert' requires immediate action and evokes a sense of urgency, while an 'FYI' message is purely informative and does not necessitate immediate action.

4.Question

What are some key practices to create good alerts?

Answer:Key practices include: stop using email for alerts, create and utilize runbooks, avoid arbitrary static thresholds, delete or tune alerts, implement maintenance periods, and attempt automated self-healing before escalating to humans.

5.Question

How can automated self-healing help reduce alert

fatigue?

Answer:Automated self-healing reduces alert fatigue by automatically resolving common issues that generate alerts, preventing unnecessary disruptions to on-call personnel.

6.Question

What are some strategies to improve on-call experiences for team members?

Answer:Strategies to improve the on-call experience include establishing clear on-call rotations, ensuring adequate time between shifts, providing compensation or PTO following on-call duties, and involving engineers in the on-call process.

7.Question

Why is incident management important after a service outage?

Answer:Incident management is crucial because it establishes a structured response to outages, ensuring that issues are logged, addressed promptly, and learned from to prevent future occurrences.

8.Question

What is a postmortem and why is it essential in incident

management?

Answer:A postmortem is a discussion held after an incident to analyze what went wrong, why it happened, and how to improve processes going forward, fostering a culture of learning rather than blame.

9.Question

What impact can a blame culture have on team performance?

Answer:A blame culture can stifle open communication, discourage honesty about mistakes, and prevent teams from addressing underlying issues, ultimately hindering improvement and learning.

10.Question

What should teams do to maintain the effectiveness of their alerting strategy?

Answer:Teams should regularly evaluate and refine their alerts based on historical data and performance, ensuring alerts are relevant and actionable without contributing to alert fatigue.

Why Bookey is must have App for Book Lovers

30min Content
The deeper and clearer interpretation we provide, the better grasp of each title you have.

Text and Audio format
Absorb knowledge even in fragmented time.

Quiz
Check whether you have mastered what you just learned.

And more
Multiple Voices & fonts, Mind Map, Quotes, IdeaClips...

Chapter 4 | 4. Statistics Primer| Q&A

1.Question

Why is statistics considered an undervalued and misunderstood topic in software engineering and systems administration?

Answer:Many professionals believe that merely applying statistics can magically resolve problems without fully understanding the principles behind it. This leads to ineffective alerting systems and missed opportunities for meaningful insights.

2.Question

What is the effect of simply relying on tools like Nagios for alerting?

Answer:Nagios makes setting alerts easy but often leads to noisy and unhelpful alerts. For instance, a single threshold breach may trigger an alert, ignoring the broader context of data trends and behaviors.

3.Question

How can decoupling data collection from alert checks enhance monitoring practices?

Answer:By collecting data over time and analyzing those metrics separately, teams can run statistical checks on historical data instead of relying on single data points, leading to more accurate problem detection.

4.Question

What is a moving average, and how is it useful in monitoring?

Answer:A moving average smooths out data by calculating averages as new datapoints are collected, making patterns easier to spot in what would otherwise be a noisy dataset, thus facilitating better trend analysis.

5.Question

In what scenarios is median a more reliable measure than mean?

Answer:The median is more reliable when dealing with skewed data sets, where extreme outliers can distort the average. For example, income data is often better represented by the median to reflect the true earnings of a population.

6.Question

How does seasonality in data affect predictions and

planning?

Answer:Recognizing seasonal patterns in data, such as fluctuations in web server requests, allows for better planning and forecasting, helping teams anticipate demand based on historical trends.

7.Question

What common pitfalls should one avoid when calculating and using percentiles?

Answer:When using percentiles, it’s crucial to remember that you can’t average them because those calculations may ignore significant portions of data. Always base percentiles on complete datasets to avoid skewed results.

8.Question

What should practitioners consider when applying standard deviation to their data?

Answer:Practitioners should be cautious with standard deviation, as it primarily applies to normally distributed datasets. If the data is not normally distributed, the standard deviation might not provide accurate insights.

9.Question

What key questions should you ask to better understand your data attributes?

Answer:Consider if the data is skewed, the commonality of outliers, and whether the data points have natural upper or lower bounds, as these factors influence the choice of statistical analysis methods.

10.Question

How can keeping historical data benefit monitoring in the long term?

Answer:Recording historical data enables teams to utilize statistics for detecting problems and trends over time, fostering a deeper understanding of systems performance and paving the way for predictive analysis.

Chapter 5 | 5. Monitoring the Business| Q&A

1.Question

What is the importance of monitoring from the user's perspective?

Answer:Monitoring from the user's perspective helps to gain immediate insight into the actual

questions people are asking, such as 'Is the site up?' and 'Are users impacted?'. This approach allows engineers to understand and prioritize the most significant issues from a business standpoint rather than getting lost in technical details.

2.Question

How can understanding business KPIs enhance a technician's contribution to the company?

Answer:By learning to ask the same questions that executives ask—like customer usage, profitability, and user satisfaction—technicians can focus their efforts on solving high-leverage problems that directly impact the business's health, thereby demonstrating the value of their monitoring work.

3.Question

Can you list some common business KPIs?

Answer:Some common KPIs include Monthly Recurring Revenue (MRR), Revenue per Customer, Customer Lifetime Value (LTV), Net Promoter Score (NPS), Customer

Acquisition Cost (CAC), Customer Churn Rate, and Active Users.

4.Question

Why is it essential to tie business KPIs to technical metrics?

Answer:Tying business KPIs to technical metrics creates a holistic view that helps to understand the health and performance of the application more deeply. It allows for the tracking of success and failure rates as well as response times, giving clear indicators of what might be going wrong with the service.

5.Question

What role do product managers play in determining which metrics to monitor?

Answer:Product managers have a comprehensive understanding of customer needs and the functionality of the app, making them ideal resources for identifying important KPIs. Their insights help in determining which aspects of the service are paramount to monitor for optimal business

performance.

6.Question

What steps should one follow to find their company's business KPIs?

Answer:To find your company's business KPIs, talk to product managers, software engineering team leads, and senior engineers to gain insights into what metrics matter most. Additionally, mapping out the app's high-level functionality helps clarify what metrics should be monitored.

7.Question

How can metrics like Customer Churn indicate issues with the application?

Answer:A high churn rate may signal potential problems such as poor product quality, performance issues, or high costs. It suggests that users are dissatisfied and may leave due to these issues, making it critical to investigate the underlying causes.

8.Question

What are the implications of not having predefined monitoring in a software system?

Answer:Not having predefined monitoring leads to a reactive approach where problems are only addressed after they arise, resulting in missed opportunities for performance enhancement and potentially severe outages. Instead, monitoring must be integrated into the design of the app from the start.

9.Question

Why is it a challenge to get access to business metrics data?

Answer:Business metrics can be sensitive in nature, leading to restricted access for technical teams. Understanding which metrics are being measured at the executive level is crucial for aligning technical monitoring efforts with business objectives.

10.Question

How does the author suggest modifying the application for better monitoring?

Answer:The author suggests that if the app doesn’t provide the necessary measurements for monitoring, teams should

take the initiative to modify the app or infrastructure to include those metrics, ensuring that they have the visibility needed to maintain performance.

Chapter 6 | 6. Frontend Monitoring| Q&A

1.Question

Why is frontend monitoring often overlooked in companies?

Answer:Frontend monitoring is frequently neglected because many view it as an area solely managed by Ops teams. This leads to a significant blind spot in performance assessment since the frontend—comprising HTML, CSS, JavaScript, and images—plays a crucial role in the user experience.

2.Question

What distinguishes frontend monitoring from backend monitoring?

Answer:Frontend monitoring focuses on the performance metrics and issues that users experience in their web browser or device, including page load times and JavaScript errors. In

contrast, backend monitoring deals with server-side operations, data fetching, and API interactions.

3.Question

What is a Single-Page Application (SPA) and why is it relevant to frontend monitoring?

Answer:An SPA is a web application that loads most resources on the client side, minimizing server requests. Its relevance to frontend monitoring arises from challenges like spikes in JavaScript errors that may not coincide with HTTP errors, complicating traditional monitoring approaches.

4.Question

What can be the business impact of slow page load times based on research?

Answer:Research indicates that a one-second delay in page load time can lead to an 11% decrease in page views, a 7% drop in conversions, and a 16% decline in customer satisfaction. For instance, Shopzilla saw a 12% revenue increase by improving their load time from 6 seconds to 1.2 seconds.

5.Question

What are effective approaches to frontend monitoring?

Answer:There are two main approaches: Real User Monitoring (RUM), which collects data based on actual user traffic, and synthetic monitoring, which simulates user interactions to test performance. RUM is particularly valuable as it reflects real-world user experiences.

6.Question

What constitutes the Document Object Model (DOM) and why is it important for monitoring?

Answer:The DOM is a tree-like representation of a webpage where each HTML element is a node. Understanding the DOM is crucial for monitoring because JavaScript manipulations can significantly affect performance, as scripts loaded synchronously can delay rendering.

7.Question

How should teams prioritize frontend performance improvements?

Answer:Teams should prioritize frontend performance improvements as essential for online profitability and user

retention. Concrete metrics, like load times, should be monitored regularly to measure performance and justify the time spent on enhancements.

8.Question

What practical steps can be taken to implement frontend monitoring in an organization?

Answer:Organizations can start by monitoring page load times using tools like Google Analytics, tracking JavaScript exceptions, and integrating performance testing tools like WebpageTest into their CI pipeline to ensure ongoing measurement and improvement.

9.Question

What key metrics should be monitored for assessing frontend performance?

Answer:Key metrics include: 'navigationStart' (the start of page request), 'domLoading' (when the DOM begins loading), 'domContentLoaded' (when all scripts have executed), and 'loadEventEnd' (when everything has fully loaded). These provide a comprehensive view of

user-perceived performance.

10.Question

How can the integration of frontend monitoring tools influence a company's development process?

Answer:Integrating frontend monitoring tools helps development teams identify performance issues early in the process, especially before deploying new features. This proactive approach helps maintain optimal performance and avoid degradation over time.

Positive feedback

Sara Scholz

tes after each book summary
erstanding but also make the
and engaging. Bookey has
ding for me.

Fantastic!!!

Masood El Toure

I'm amazed by the variety of books and languages Bookey supports. It's not just an app, it's a gateway to global knowledge. Plus, earning points for charity is a big plus!

Fi
Ab
bo
to
m

José Botín

ding habit
p's design
ual growth

Love it!

Wonnie Tappkx

Bookey offers me time to go through the important parts of a book. It also gives me enough idea whether or not I should purchase the whole book version or not! It is easy to use!

Time saver!

Bookey is my go-to app for
summaries are concise, ins
curated. It's like having acc
right at my fingertips!

Awesome app!

Rahul Malviya

I love audiobooks but don't always have time to listen to the entire book! bookey allows me to get a summary of the highlights of the book I'm interested in!!! What a great concept !!!highly recommended!

Beautiful App

Alex Walk

This app is a lifesaver for book lovers with busy schedules. The summaries are spot on, and the mind maps help reinforce wh I've learned. Highly recommend!

Chapter 7 | 7. Application Monitoring| Q&A

1.Question

Why is visibility into application performance crucial for organizations?

Answer:Visibility into application performance is crucial because applications are frequently modified, and these changes can directly impact user experience and business outcomes. Understanding how applications perform allows teams to transition from reactive to proactive management, ensuring any issues are addressed before they affect users.

2.Question

What common fears do teams have about application monitoring?

Answer:Teams often fear that application monitoring is too difficult or requires specialized skills. However, the chapter emphasizes that with the right tools and strategies, such as starting simple with basic performance metrics, application monitoring is accessible and incredibly beneficial.

3.Question

How can instrumenting your applications lead to proactive maintenance?

Answer:Instrumenting applications allows teams to collect valuable metrics about performance, such as response times for database queries and API calls. With this data, teams can identify potential issues before they impact users, thus moving towards proactive maintenance instead of constantly reacting to problems.

4.Question

What are the limitations of APM tools as mentioned in the chapter?

Answer:APM tools often provide detailed performance data, but they can lack the context needed to interpret that data meaningfully. They may present metrics like latency without understanding the specific workflows in which those latencies occur, making it challenging to derive actionable insights.

5.Question

How does StatsD simplify application performance

monitoring?

Answer:StatsD simplifies application performance monitoring by allowing developers to add metrics to their applications with minimal effort. Its ease of use makes it a staple in modern monitoring stacks, enabling teams to track important metrics effectively without complex setups.

6.Question

What is the purpose of a health endpoint in applications?

Answer:A health endpoint in applications provides a simple way to check the application's operational status. It tells whether the application is healthy, detailing its dependencies and their statuses, which aids in quick diagnosis and alerting in case of issues.

7.Question

Why is it important to monitor deployment pipelines?

Answer:Monitoring deployment pipelines provides insights into the build and deployment process, helping teams detect regressions and linking performance metrics to specific releases. This correlation can uncover the root causes of

performance issues that arise after new code is deployed.

8.Question

What strategy does the author suggest for logging in applications?

Answer:The author suggests using structured logs over unstructured ones to enhance data retrieval and analysis. By structuring logs, teams can ensure that logs capture all relevant context and metadata, aiding more efficient troubleshooting and performance monitoring.

9.Question

What are two rules of thumb to consider when deciding between metrics and logs?

Answer:The two rules of thumb are: 1) Consider whether it is easier for your team to think about metrics or logs for the specific use case; 2) Evaluate whether the information in question would be more effective as a log entry or a metric.

10.Question

What should teams be cautious about regarding log levels?

Answer:Teams should be cautious that relying too heavily on

log severity levels can lead to missing critical debugging information. If logging levels are set incorrectly, essential data may not be captured when problems occur, hindering troubleshooting efforts.

Chapter 8 | 8. Server Monitoring| Q&A

1.Question

What is the primary misconception about server monitoring that the author addresses?

Answer:Many associate server monitoring solely with what sysadmins do, overlooking the broader aspects of monitoring beyond servers.

2.Question

Why should standard OS metrics like CPU and memory not be the primary focus for alerting?

Answer:Starting with standard metrics offers the least signal regarding application performance; instead, focus on metrics that directly indicate application health.

3.Question

What is an effective way to use standard OS metrics according to the author?

Answer:Automatically record standard OS metrics for diagnostics and troubleshooting but avoid setting alerts unless there's a compelling reason.

4.Question

What crucial elements should be monitored in CPU usage?

Answer:Monitor CPU utilization percentages and ensure that it reflects the actual work being performed, particularly focusing on user and system processes.

5.Question

How should memory usage be interpreted to determine if more memory is needed?

Answer:Focus on memory usage reported without buffers and cache, as they are technically available for use, to accurately assess whether memory is genuinely under pressure.

6.Question

What log entries should be monitored to detect memory issues?

Answer:Track occurrences of the OOMKiller in logs, as it

indicates serious memory pressure by terminating processes to free up memory.

7.Question

What are some essential metrics to monitor for network performance?

Answer:Collect metrics on octets in/out, errors, and drops on your server interfaces to gauge network performance effectively.

8.Question

Why should IOPS (Input/output operations per second) be monitored closely for disk performance?

Answer:IOPS is critical for identifying and diagnosing performance issues in database servers or any service reliant on disk performance.

9.Question

What is a recommended practice for monitoring SSL certificates?

Answer:Set up proactive monitoring for SSL certificate expiration through domain registrars or external monitoring tools to avoid lapses.

10.Question

Why does the author advise against using SNMP for server monitoring?

Answer:SNMP is complex to configure, can create security vulnerabilities, and there are more efficient and secure alternatives available for monitoring.

11.Question

What is the best practice for monitoring web server performance?

Answer:Track requests per second (req/sec) as the golden metric for web server performance, alongside monitoring HTTP response codes for visibility.

12.Question

What metric indicates how busy a database server is?

Answer:Monitor queries per second (qps) to understand how active and loaded your database server is, beyond just connection counts.

13.Question

What is a "dead man's switch" and how can it be implemented for monitoring scheduled jobs?

Answer:A dead man's switch triggers alerts when expected actions do not occur within a set timeframe, helping catch failures of scheduled cron jobs.

14.Question

What should be prioritized regarding logging practices?

Answer:Focus on log collection, sending them to a centralized, searchable log management solution, and conducting thorough log analysis for actionable insights.

15.Question

How can I prevent missing important log entries when using syslog for log collection?

Answer:Use TCP for syslog forwarding to ensure reliable delivery of logs and prevent loss of critical entries.

16.Question

What are actionable items the author suggests for improving monitoring strategies?

Answer:Implement structured monitoring of various components, such as load balancers, message queues, caching systems, DNS, and keeping track of potential issues as structured data for better analysis.

Chapter 9 | 9. Network Monitoring| Q&A

1.Question

Why is network monitoring crucial for network reliability?

Answer:Network monitoring is fundamental because the behavior and performance of a network directly affect everything that relies on it. Issues like downtime or slow performance can lead to widespread disruptions affecting users' access to applications and services.

2.Question

What was the author's personal experience that sparked their interest in network monitoring?

Answer:The author became interested in network monitoring after accidentally disconnecting a network switch at their workplace, which resulted in numerous employees being unable to check their email until the issue was resolved the next day.

3.Question

How does the author describe the importance of

understanding and using SNMP in network monitoring?

Answer:The author describes SNMP as an essential but often arcane protocol for network monitoring, indicating that while it can feel outdated, it is crucial for retrieving information from network devices effectively.

4.Question

What are the drawbacks of SNMP as mentioned in the chapter?

Answer:SNMP is considered insecure since it passes community strings and data in plain text. Moreover, it can be complex to implement due to variations in how different vendors implement the protocol and its inherent limitations.

5.Question

What fundamental metrics should you monitor in networking?

Answer:Key metrics to monitor include bandwidth, throughput, latency, errors, and jitter. Understanding these can help you identify performance issues in your network.

6.Question

Can you explain the difference between bandwidth and

throughput?

Answer:Bandwidth is the theoretical maximum speed at which data can be transmitted over a network link, expressed in bits per second, while throughput is the actual observed performance during transmission, which tends to be lower due to various overheads.

7.Question

Why is capacity planning important, and how can network monitoring assist with it?

Answer:Capacity planning is crucial for anticipating the network's future needs based on current usage trends. Network monitoring provides the data necessary to identify when to upgrade hardware or increase bandwidth to meet growing demands.

8.Question

What is the author's perspective on monitoring configuration changes in network devices?

Answer:The author emphasizes that tracking configuration changes is vital for troubleshooting and understanding

incidents within the network, as many issues arise from untracked changes made by staff.

9.Question

What can voice and video performance monitoring reveal?

Answer:Voice and video monitors reveal performance issues such as latency and jitter, which are critical for maintaining the quality of streams, ensuring that they remain stable and deliver a satisfactory user experience.

10.Question

What challenges do users face when monitoring routing protocols?

Answer:Dynamic routing protocols like BGP and OSPF present challenges because they are designed to self-correct, making it less clear when issues arise that warrant alerts.

11.Question

How does the author suggest monitoring can improve troubleshooting in network environments?

Answer:By establishing a framework of metrics to track, like interface performance and error rates, network admins can

more effectively diagnose issues, correlate changes, and respond to specific incidents rapidly.

Read, Share, Empower

Finish Your Reading Challenge, Donate Books to African Children.

The Concept

BOOKS FOR AFRICA × ×

This book donation activity is rolling out together with Books For Africa. We release this project because we share the same belief as BFA: For many children in Africa, the gift of books truly is a gift of hope.

The Rule

Earn 100 points → **Redeem a book** → **Donate to Africa**

Your learning not only brings knowledge but also allows you to earn points for charitable causes! For every 100 points you earn, a book will be donated to Africa.

Chapter 10 | 10. Security Monitoring| Q&A

1.Question

How does security monitoring differ from infrastructure or application monitoring?

Answer:Security monitoring requires a proactive approach to identify and manage threats, while infrastructure monitoring focuses on the existing systems and their performance metrics.

2.Question

What does the author mean by security being a continuum?

Answer:Security is a spectrum where at one end is very low security (like a 'wet paper bag') and at the other end is extremely high security (like 'Fort Knox'). You need to assess the level of threat and risk to determine the appropriate level of security.

3.Question

Why is it important to consider compliance regulations in security monitoring?

Answer:Compliance regulations mandate that organizations

demonstrate control over their data and processes. Monitoring plays a crucial role in ensuring that these controls are working effectively.

4.Question

What role does auditd play in security monitoring?

Answer:Auditd is a userspace interface to the Linux Audit System, which enables the tracking of user actions and system events for security purposes.

5.Question

How can organizations effectively monitor for compliance?

Answer:Organizations should implement monitoring for all components affected by the compliance regulations to demonstrate that their controls are functioning as intended.

6.Question

What are some of the challenges faced when implementing security monitoring tools?

Answer:Challenges include the potential intrusiveness of security measures on workflows, the cost of implementation, and difficulty in integrating security awareness into existing

systems.

7.Question

What is the significance of using remote logs in security monitoring?

Answer:Using remote logs ensures that audit logs are stored securely and are less vulnerable to tampering, allowing for reliable analysis and incident response.

8.Question

How does a NIDS complement other security measures?

Answer:A Network Intrusion Detection System (NIDS) identifies and alerts on network threats after they occur, providing essential information on breaches even when preventive defenses fail.

9.Question

What should you consider when deploying network taps for a NIDS?

Answer:Strategic placement of network taps at choke points in the network is essential to capture as much traffic as possible for analysis, ensuring no critical data is missed.

10.Question

Why is tuning important for monitoring tools like NIDS?
Answer:Regular tuning of monitoring tools ensures that they effectively filter out noise and accurately report on genuine threats, improving the overall security posture.

11.Question

What is a key takeaway about the importance of security monitoring according to the author?
Answer:Effective security monitoring is essential for protecting applications and infrastructures from threats, and it requires a comprehensive understanding of compliance needs and proactive measures.

Chapter 11 | 11. Conducting a Monitoring Assessment| Q&A

1.Question

What is the purpose of conducting a monitoring assessment in a business environment?
Answer:Conducting a monitoring assessment systematically determines what needs to be monitored and why. This results in a clearer understanding of application behavior and the

underlying infrastructure.

2.Question

How does Tater.ly make money from its service?

Answer:Tater.ly generates revenue through advertising, specifically by featuring ads for restaurants at the top of search results for a fee based on the number of impressions.

3.Question

What are some key performance indicators (KPIs) important for Tater.ly?

Answer:Important KPIs for Tater.ly include the number of restaurants reviewed, active restaurant owners, number of users, active users, searches performed, reviews placed, and ads purchased. Additionally, net promoter scores from both users and restaurants are critical.

4.Question

What types of metrics should be monitored on the frontend?

Answer:Frontend monitoring should focus on Real User Monitoring (RUM) metrics to track page load times and user interactions from the user's perspective.

5.Question

In addition to basic metrics, what specific metrics should be monitored for Tater.ly's backend?

Answer:Backend metrics should include page load time, user logins (successes/failures), daily/weekly active users, search latency, review submissions latency, PostgreSQL transaction rates, Redis hit/miss ratios, CPU and memory utilization, and network throughput.

6.Question

How should Tater.ly approach security monitoring?

Answer:Tater.ly should monitor SSH login attempts and failures, syslog logs, and audit logs to ensure security, even without compliance requirements.

7.Question

What alerts should be configured based on the identified metrics and logs?

Answer:Alerts should be set for increasing page load times, error rates, latency on critical components (Redis, Apache), and significant latency on PostgreSQL queries.

8.Question

What is the significance of this monitoring assessment for Tater.ly?

Answer:This assessment helps Tater.ly establish a foundational monitoring strategy that will evolve alongside the business and technological landscape, ensuring adaptive and effective operations.

9.Question

What lesson can be taken from the conclusion of the monitoring assessment?

Answer:The key takeaway is that monitoring is an ongoing process; the assessment serves as a starting point, encouraging continuous improvement as the business and infrastructure develop.

10.Question

Why is it important to write a runbook after completing a monitoring assessment?

Answer:Writing a runbook helps document the knowledge gained from the assessment, ensuring that colleagues can utilize this information for operational benefits and effective

incident management.

Chapter 12 | A. An Example Runbook: Demo App| Q&A

1.Question

What is the purpose of using a runbook in monitoring operational activities?

Answer:A runbook serves as a structured guideline for handling operational tasks and incidents. It consolidates essential information and procedures to ensure that team members can respond effectively to issues. By starting with a basic template, like the example provided, teams can customize runbooks over time to suit their specific needs, enhancing efficiency and accuracy in incident management.

2.Question

How should a runbook be approached according to this chapter?

Answer:Runbooks should be viewed as living documents that evolve over time. The chapter encourages teams to build on the initial example, iterating and updating the content as

they learn and adapt to new insights, challenges, or changes in the infrastructure. A runbook is only as useful as the information it contains, so constant refinement is key.

3.Question

Why is it important to document external and internal dependencies in a runbook?

Answer:Documenting dependencies in a runbook is crucial as it helps the team understand what systems and components that their application relies on. This knowledge aids in troubleshooting, ensuring that problems can be quickly identified and addressed, particularly during incidents where external or internal systems may fail or behave unexpectedly.

4.Question

What types of metrics and logs are essential for the Demo App, and why?

Answer:The Demo App emits critical metrics such as user login counts, post creation times, and more. These metrics allow teams to gauge performance and user engagement.

Logging key events, like user sign-ins and actions taken with posts and comments, supports the identification of success rates and failure patterns, which are vital for monitoring system health and for troubleshooting purposes.

5.Question

What should be done when an alert is triggered, according to the alerting guidelines?

Answer:When an alert is triggered, immediate investigation is necessary. For example, if there is a high user signin failure rate, check for recent deployments that may have introduced issues or scan the signin logs for signs of a brute force attack. This proactive approach helps to mitigate potential problems before they escalate.

6.Question

How can alerts assist in maintaining application performance?

Answer:Alerts provide immediate feedback when performance metrics exceed designated thresholds, such as login times or post creation times. By monitoring these

parameters, teams can quickly identify and troubleshoot issues like bottlenecks or deploy problems. This proactive monitoring helps maintain user satisfaction and system reliability.

7.Question

What does the example escalation procedure highlight about communication during incidents?

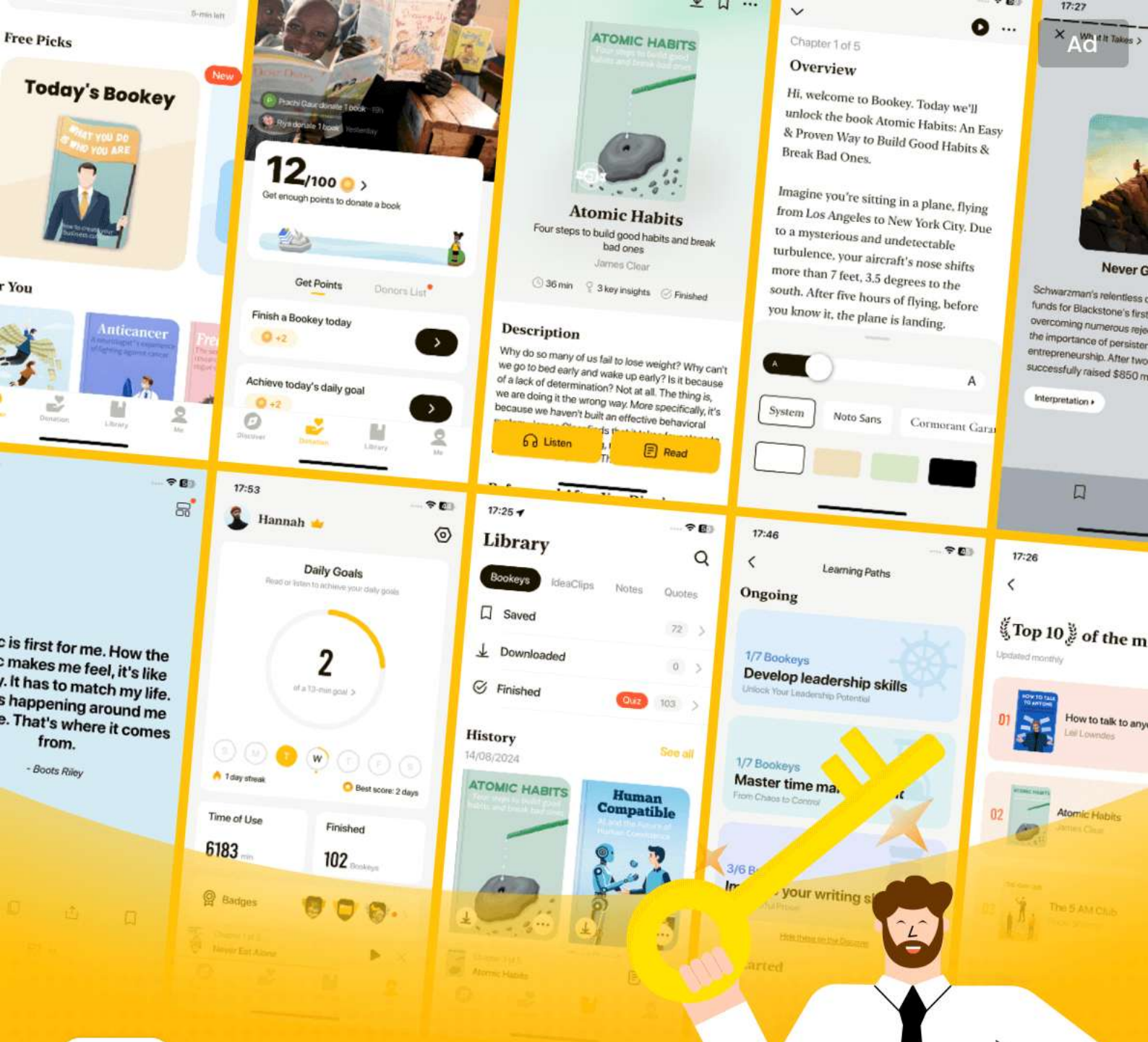
Answer:The escalation procedure emphasizes the importance of clearly defined contact points within a team. It ensures that if an issue arises that can't be handled at the first level, there’s a straightforward way to involve the service owner or other responsible parties. Effective communication prevents confusion and enables faster resolution of incidents.

8.Question

What does the chapter imply about the relationship between runbooks and team learning?

Answer:The chapter implies that runbooks are not only tools for documenting procedures but also instruments for team growth and learning. As teams encounter new challenges and

collect experiences, they should continually refine their runbooks to incorporate lessons learned. This iterative process bolsters team knowledge and prepares them for future incidents.



World' best ideas unlock your potential

Free Trial with Bookey

Download on the App Store

GET IT ON Google Play

Scan to download

Chapter 13 | B. Availability Chart| Q&A

1.Question

What is the significance of understanding downtime in relation to availability percentages?

Answer:Understanding the relationship between downtime and availability percentages is crucial for businesses as it allows them to set realistic uptime targets that align with operational requirements and customer expectations. For example, if a service is set to maintain 99.9% availability, this translates to only about 8.76 hours of downtime per year, which is vital information for teams to strategize accordingly and ensure they do not exceed this limit.

2.Question

How does the concept of 'nines' in availability help organizations gauge their performance?

Answer:The 'nines' refer to the level of availability, such as 99.9% (three nines), which indicates how reliable a service or system is. This concept helps organizations gauge

performance by providing a concrete measure of uptime. A company striving for five nines (99.999%) is expected to have only about 5.26 minutes of downtime annually, which requires robust strategies for reliability and incident response to meet such stringent targets.

3.Question

How might a team utilize the availability chart in planning maintenance?

Answer:A team can use the availability chart to plan maintenance windows strategically. For instance, if aiming for a 99.5% availability target, knowing they can only afford 1.83 days of downtime per year means they must carefully anticipate and schedule maintenance events to avoid exceeding this limit. This approach ensures that customers experience minimal disruption and that service level agreements are honored.

4.Question

Can you elaborate on what happens when a service does not meet its availability target?

Answer:When a service fails to meet its availability target, it can lead to significant repercussions, such as lost revenue, customer dissatisfaction, and potential damage to the brand reputation. For instance, if a company promises 99.99% uptime but experiences more downtime than allowed (over 52.56 minutes per year), this can frustrate customers who rely on that service, possibly causing them to seek alternatives.

5.Question

What are the implications of achieving levels such as 99.9999999% ('nine nines') availability?

Answer:Achieving 'nine nines' availability (99.9999999%) is exceedingly rare and generally indicates a system designed with extreme redundancy, highly redundant systems, and an impeccable incident response plan. For organizations, it may be over-engineering and economically infeasible for standard operations, but it sets a benchmark for reliability that can inspire teams to innovate and improve even within their more realistic targets.

Practical Monitoring Quiz and Test

Check the Correct Answer on Bookey Website

Chapter 1 | 1. Monitoring Anti-Patterns| Quiz and Test

1. Focusing on acquiring new monitoring tools is the most effective way to solve monitoring problems.
2. Every team member should participate in the monitoring process to improve system reliability.
3. The concept of a 'Single Pane of Glass' in monitoring is a realistic and achievable goal.

Chapter 2 | 2. Monitoring Design Patterns| Quiz and Test

1. Composable monitoring encourages using monolithic designs for monitoring systems.
2. Visualization techniques are crucial for operational efficiency in monitoring systems.
3. Julian suggests that companies should rush to achieve advanced monitoring levels immediately.

Chapter 3 | 3. Alerts, On-Call, and Incident Management| Quiz and Test

1. Alerts are primarily used for merely sending notifications without observing system behavior over time.
2. A good alert should evoke urgency and require immediate action.
3. Using email for alerts is recommended to prevent alert fatigue and maintain human attention.

Download Bookey App to enjoy

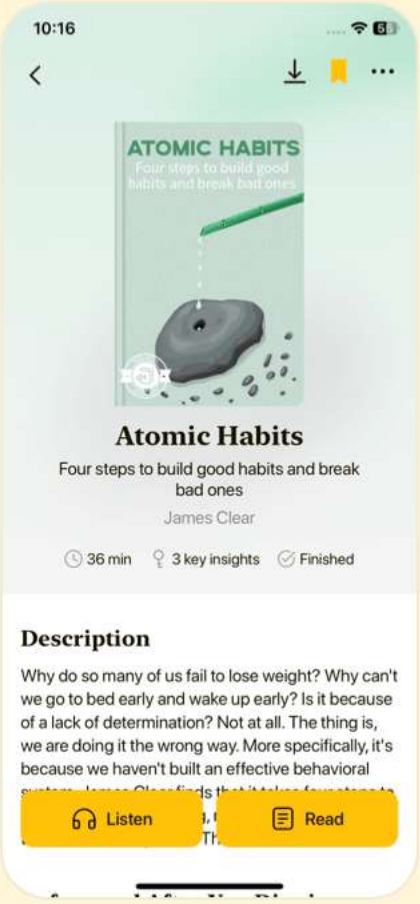
1000+ Book Summaries with Quizzes

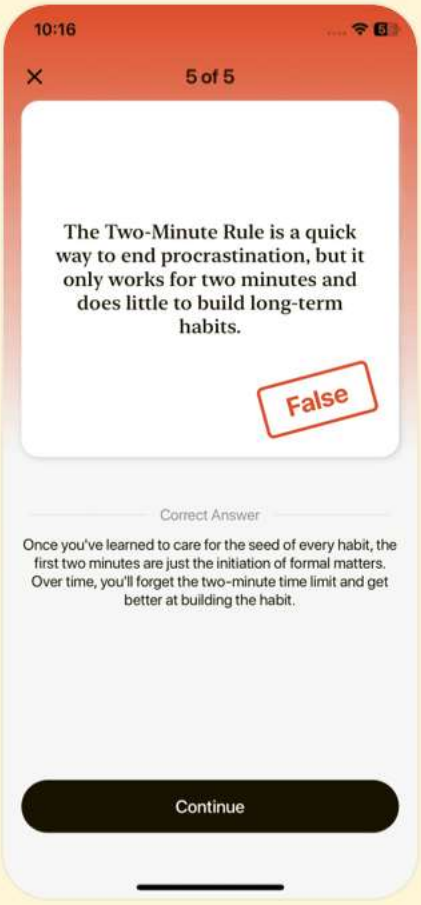
Free Trial Available!

Scan to Download

Download on the App Store

GET IT ON Google Play





Chapter 4 | 4. Statistics Primer| Quiz and Test

1. Statistics is often undervalued and misunderstood in software engineering and systems administration.
2. Standard deviation is a reliable metric for all datasets regardless of their distribution.
3. The mean and median are always the same and can be used interchangeably when analyzing data.

Chapter 5 | 5. Monitoring the Business| Quiz and Test

1. Chapter 5 emphasizes the importance of monitoring from a technical perspective rather than a business perspective.
2. Common questions from executives include customer usability, profitability, and growth trends.
3. To find relevant KPIs, it is unnecessary to engage with product managers and engineering teams.

Chapter 6 | 6. Frontend Monitoring| Quiz and Test

1. Frontend monitoring is often overlooked by

companies, considering it the sole responsibility of operations.

2. Single-Page Applications (SPAs) load most resources server-side, leading to more server requests and faster performance.
3. Real User Monitoring (RUM) should be the cornerstone of frontend monitoring as it reflects real-world user experiences.

Download Bookey App to enjoy

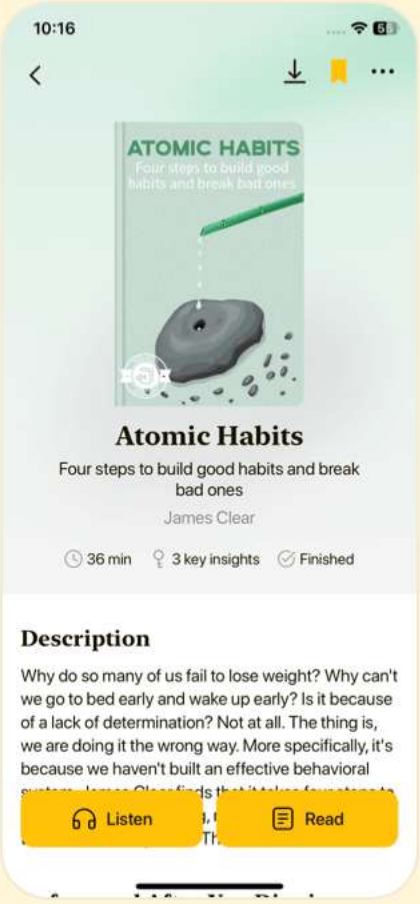
1000+ Book Summaries with Quizzes

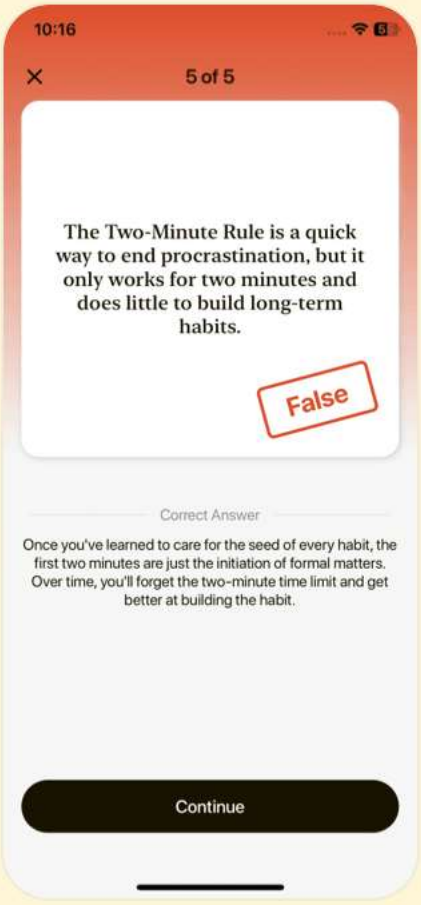
Free Trial Available!

Scan to Download

Download on the App Store

GET IT ON Google Play





Chapter 7 | 7. Application Monitoring| Quiz and Test

1. Most organizations excel in monitoring applications and overlook server infrastructure monitoring.
2. APM tools automatically gather data about application performance and provide full context about application behavior.
3. Structured logs, particularly in JSON format, are recommended for clarity and ease of analysis in application logging.

Chapter 8 | 8. Server Monitoring| Quiz and Test

1. Monitoring is solely the responsibility of system administrators and does not involve other roles.
2. Standard OS metrics like CPU and memory should be the primary focus for setting alerts according to the chapter.
3. Monitoring SSL certificates is important in order to prevent expiration issues.

Chapter 9 | 9. Network Monitoring| Quiz and Test

1. Network monitoring is essential for the performance of all dependent applications.
2. SNMP has not changed since its introduction in 1988, making it challenging for modern use.
3. Tracking changes in network device configurations is vital for understanding incidents and maintaining network stability.

Download Bookey App to enjoy

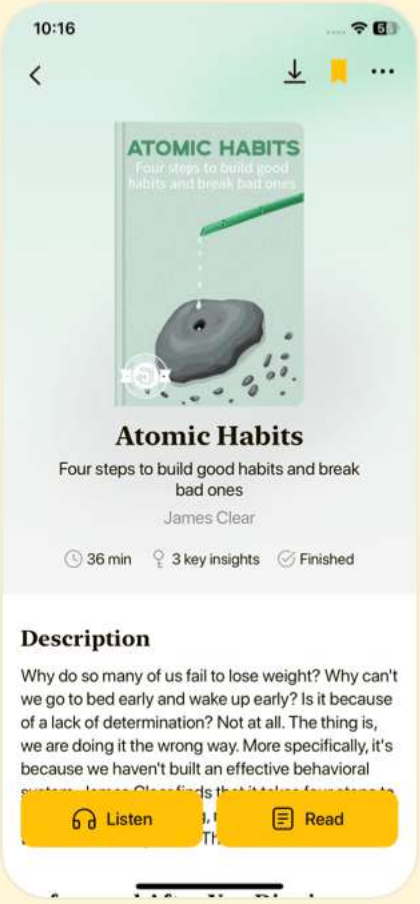
1000+ Book Summaries with Quizzes

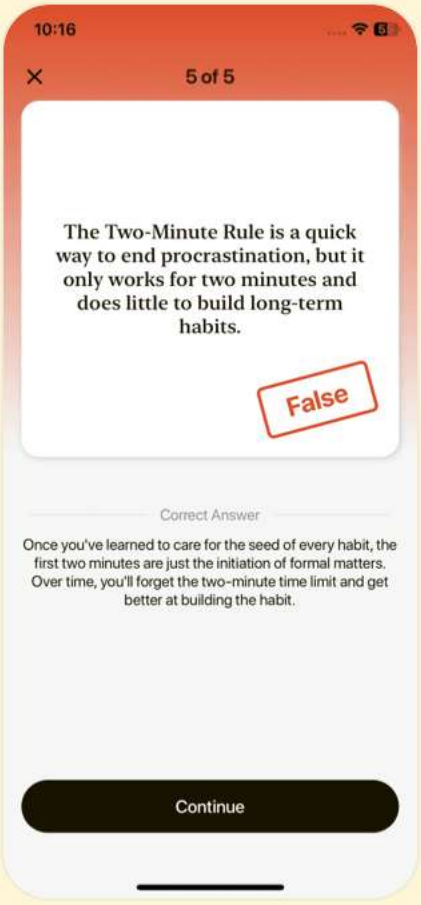
Free Trial Available!

Scan to Download

Download on the App Store

GET IT ON Google Play





Chapter 10 | 10. Security Monitoring| Quiz and Test

1. Security monitoring differs significantly from traditional infrastructure or application monitoring.
2. Compliance with regulations does not require monitoring various security controls.
3. auditd can be configured to track security-focused events like sudo executions and file accesses.

Chapter 11 | 11. Conducting a Monitoring Assessment| Quiz and Test

1. The purpose of conducting a monitoring assessment is to improve business KPIs for Tater.ly.
2. Frontend Monitoring should focus solely on backend server performance metrics and ignore user experience metrics.
3. Security Monitoring for Tater.ly includes monitoring SSH login attempts and is heavily based on compliance requirements.

Chapter 12 | A. An Example Runbook: Demo App|

Quiz and Test

1. A well-structured runbook is not essential for effective monitoring and incident response.
2. The demo app operates under the ownership of John Doe.
3. The demo app has multiple external dependencies that need monitoring.

Download Bookey App to enjoy

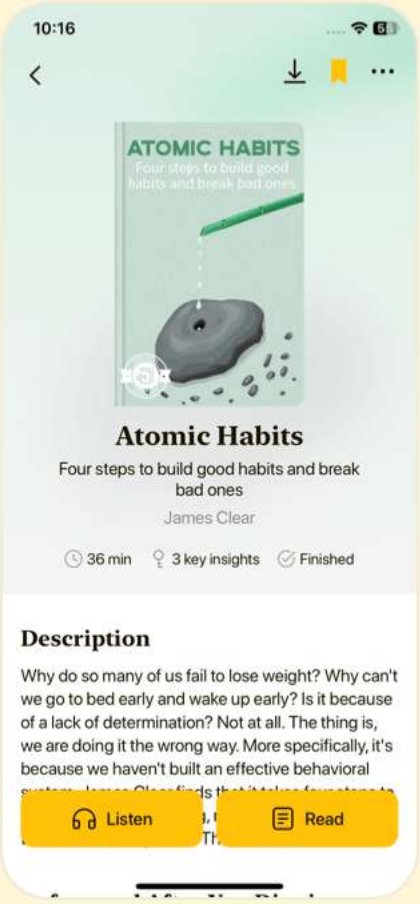
1000+ Book Summaries with Quizzes

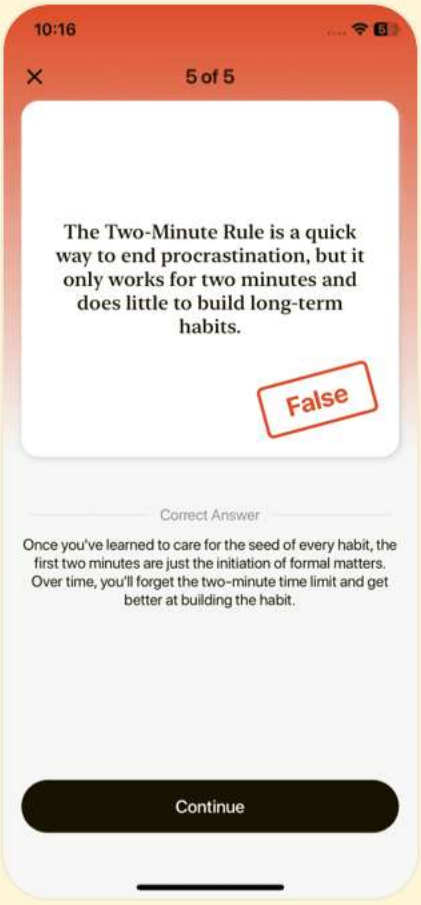
Free Trial Available!

Scan to Download

Download on the App Store

GET IT ON Google Play





Chapter 13 | B. Availability Chart| Quiz and Test

1. At 99% availability, the allowed downtime per year is 3.65 days.
2. If a system has 99.99999% availability, the downtime per week is 60.48 milliseconds.
3. The downtime per month for a 95% availability system is 36 hours.

Download Bookey App to enjoy

1000+ Book Summaries with Quizzes

Free Trial Available!

Scan to Download

Download on the App Store

GET IT ON Google Play