aws marketplace

Why Continuous Monitoring Is Important to DevOps Maturity

How to implement continuous monitoring on Amazon Web Services (AWS)

Introduction

Customers today expect applications to be reliable, seamless, and "always on." To accomplish this, organizations must be quick and agile about delivering mature products at the speed of the market. Continuous monitoring is a holistic approach that provides automated visibility into your entire software delivery pipeline and the operations and infrastructure that it depends on. In this whitepaper, DevOps Institute Ambassador, Marc Hornbeek explains why well-engineered, mature continuous monitoring is important to improve failure detection, analysis, prediction, and response. He also offers a self-assessment tool for helping organizations measure their level of continuous monitoring maturity.

Building on Hornbeek's perspective, AWS Marketplace will share how you can specifically apply this process to your AWS environment. You will be introduced to relevant AWS Marketplace solutions that can enhance your continuous monitoring strategy. Finally, PagerDuty will be featured as an available option for strengthening your continuous monitoring in AWS.

Author: **Marc Hornbeek**
Ambassador – DevOps Institute
CEO – Engineering DevOps Consulting
Analyst – Accelerated Strategies Group
Author – "Engineering DevOps"

aws marketplace

Continuous Monitoring Maturity

As the name implies, *Continuous Monitoring* refers to the capability for using software and networks that connect sources of monitoring data with tools that collect and analyze the data to provide actionable responses to incidents, as automated as possible.

> *"Continuous Monitoring provides automated visibility of the health of all things that are important to the operation and performance of the application, the Continuous Delivery pipeline, and the infrastructure that they depend on."*
>
> **Book Reference "Engineering DevOps" by Marc Hornbeek**

The health of DevOps affects the performance of Continuous Delivery, including leadership and culture. Continuous Monitoring is an essential capability that enables collaboration across IT specialties, reducing or eliminating the finger-pointing and other unproductive behaviors that show up all too often in organizations with operational silos. Continuous Monitoring is different than conventional monitoring. It requires taking a holistic view of the complex application environment. Continuous Monitoring requires organizational alignment. Development, QA/Test, and Operations must agree upon an implementation strategy that allows Continuous Monitoring to become a source of truth for the entire organization.

Why Is Continuous Monitoring Maturity Important to DevOps?

With DevOps, the rate of request for change increases, and with that comes an increase of risk. Continuous Monitoring offers a means to control the risk and provides a feedback mechanism for the health of the application release, the pipeline, and the infrastructure. The following are example benefits of well-engineered Continuous Monitoring:

- End-to-end visibility into your applications and database performance
- Continuous feedback loops to support continuous improvements
- Continuous Monitoring of the End-User-Experience (EUE) providing customer feedback essential to identifying product preferences and improvements
- Early detection of performance bottlenecks across the application, pipeline, and infrastructure
- Higher confidence in achieving SLAs in production
- Better prediction of code behavior
- Risk management
- Cost management
- Performance management
- Application complexity management
- Better management of the Digital Experience.

How Is Continuous Monitoring Engineered for DevOps?

Figure 1—DevOps Continuous Monitoring Blueprint shows how mature Continuous Monitoring systems are configured.

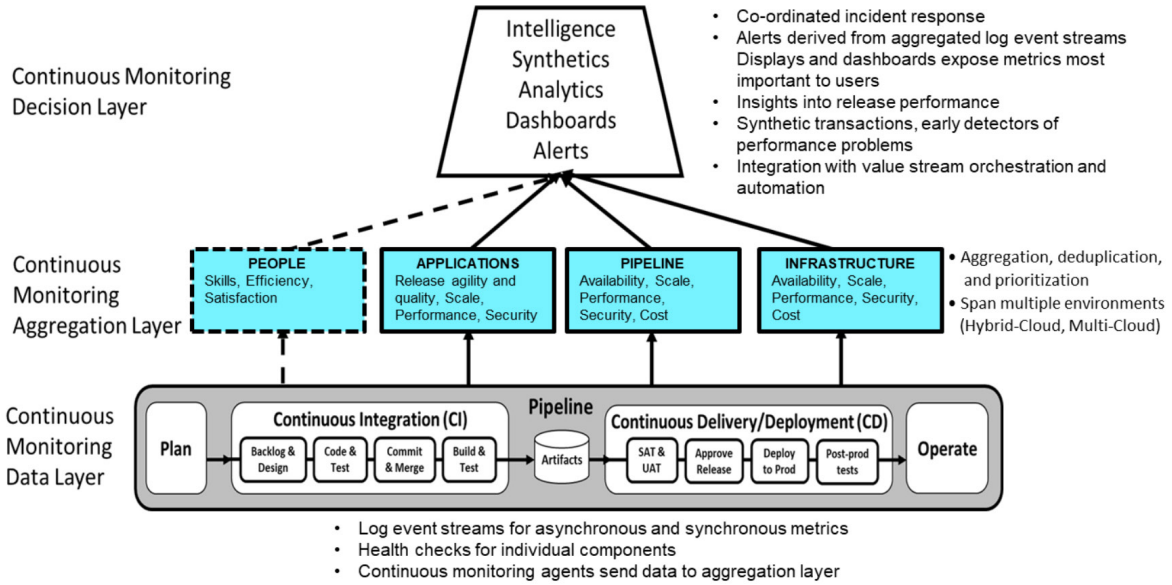

Figure 1—DevOps Continuous Monitoring Blueprint

As illustrated in *Figure 1—DevOps Continuous Monitoring Blueprint*, a key challenge for Continuous Monitoring Engineering is the selection of data to be monitored, how to aggregate it, and how to implement decisions based on the aggregated data.

The Continuous Monitoring Data Layer includes, for example, the following:

- Log event streams for asynchronous and synchronous metrics needed to assess operational performance deviations from applications, pipelines, and infrastructure components (Notifications intended to be read by a human are pushed to a ticket queue, email, or pager). Alert systems may offer users subscription services to tailor their alerts to their specific roles. Examples of alert information include reactive alerts.
- Health checks for individual components
- Continuous Monitoring agents positioned with the deployed code act as event collectors to send to an event aggregation server

The Continuous Monitoring Aggregation Layer includes, for example, the following:

- Aggregation, deduplication, and prioritization of logged event streams
- Ability to span multiple environments (i.e., Cloud, Hybrid-Cloud, Multi-Cloud, On-Premise)

The Continuous Monitoring Decision Layer includes, for example, the following:

- Alerts that are derived from aggregated log event streams
- Displays and dashboards, usually web-based, providing a visual summary built to expose metrics that are most important to users (Dashboards may include selectors, filters, and telescoping features that allow users to refine the view. Examples of items to display include current values, trend graphs, queue lengths, priority, and owners).
- Release comparison giving clear insight into release performance
- Synthetic transactions, which can be early detectors of performance problems

Processes, procedures, and policies that govern "how" you will operate and "who" will operate critical functions must be defined. This requires aligning your business goals across your entire organization to ensure that Continuous Integration and delivery are achieved. DevOps requires collaborative teams. It is perhaps the most critical success factor for high-performance DevOps. Yet collaboration is often overlooked when engineering monitoring. **Monitoring people in a constructive positive way is critical.** Metrics that emphasize reinforcement of positive behaviors instead of punitive metrics are most recommended. Some of the things that can be monitored are listed as follows.

1. Work backlog trends
2. Percent of commits successful
3. Percent of work that is creative versus corrective
4. Percent of SRE time in development
5. Job satisfaction ratings

Many application performance bottlenecks originate in the database, but too often stakeholders have little or no visibility into database performance. Here are some items that should be monitored for databases:

- Database size
- Alert when database thresholds are violated
- Trend resource consumption, database objects, schema statistics
- Database growth
- Active user counts
- Response time
- Calls and errors

The following are examples of popular tools used for Continuous Monitoring solutions:

- Intelligent analytics: Tableau, PagerDuty
- Value stream analytics: Plutora, CollabNet, VersionOne, Tasktop
- Application release metrics and analytics: CloudBees, ElectricFlow, XebiaLabs, CA Technologies, PagerDuty
- Dashboard tools: Prometheus, Hygieia (Capital One), DataDog, Service Now, PagerDuty
- Application performance monitoring tools: Dynatrace, App Dynamics, New Relic
- Infrastructure monitoring, logs, and alert tools: Nagios, Splunk, Riverbed

Continuous Monitoring Maturity Levels

There is no standard Continuous Monitoring maturity model. As a practical matter, I have found it useful to apply a five-level maturity model adapted from the "standard" software Capability Maturity Model. I define five levels of Continuous Monitoring maturity as described in the following paragraphs.

Continuous Monitoring Maturity Level 1: Chaos

Figure 2 shows key characteristics of People, Process, and Technology evident at this level of maturity. At this level there is little evidence of Continuous Monitoring skills. Pipeline processes are not well integrated and pipeline monitoring has major gaps. Frequently pipelines, applications, and infrastructure failures are first reported by unhappy users rather than monitoring systems. Typical outcomes are releases are unpredictable, reactive, and the pipeline is wasteful. Response to incidents is slow.

Figure 2—Continuous Monitoring Maturity Level 1: Chaos

Continuous Monitoring Maturity Level 2: Continuous Integration

Figure 3 shows key characteristics of People, Process, and Technology evident at this level of maturity. At this level there is some knowledge of Continuous Monitoring. The builds, integrations, and build tests are well supported by version management, and build monitoring processes and technology. End-to-end continuous delivery, especially delivery and deployment stages are not well supported with monitoring. Build quality is good, but typically there are deployment quality problems, and the deployment processes infrastructure monitoring systems have major gaps. The monitoring processes tend to have insufficient fidelity to indicate the source of failures unambiguously. This results in a situation where reaction to unknown sources of failure is chaotic. End-to-End pipeline failure events must be correlated manually.

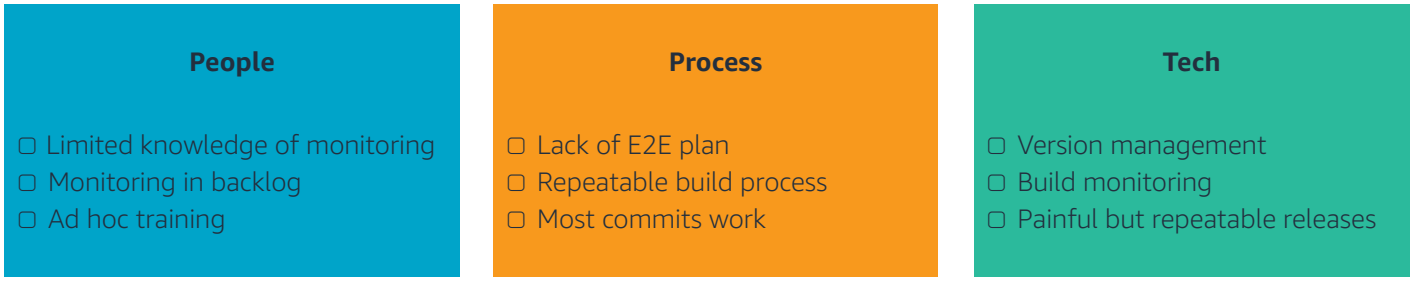

Figure 3—Continuous Monitoring Maturity Level 2: Continuous Integration

Continuous Monitoring Maturity Level 3: Continuous Flow

Figure 4 shows key characteristics of People, Process, and Technology evident at this level of maturity. At this level Continuous Monitoring processes extend from end-to-end across the pipeline. Release standards are using automated metrics and some sophisticated measures correlating application performance to business risk. Typical outcomes include repeatable quality releases with some bottlenecks and inefficiencies. There is sufficient monitoring to identify the most likely source of failure. Pro-active, monitoring sources are integrated into one place for analysis.

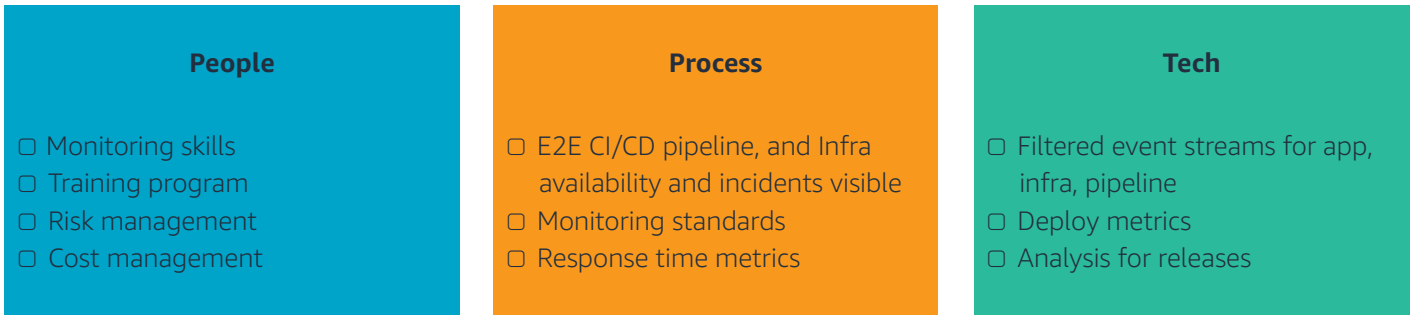

Figure 4—Continuous Monitoring Maturity Level 3: Continuous Flow

Continuous Monitoring Maturity Level 4: Continuous Feedback

Figure 5 shows key characteristics of People, Process, and Technology evident at this level of maturity. At this level more advanced knowledge of Continuous Monitoring is apparent. Goals and metrics are set for each stage in the pipeline. The culture includes training and mentoring for Continuous Monitoring. There is a focus on end-to-end performance monitoring trends rather than spot results. Automation is applied to analytics. Typical Outcomes at this level reflect a general confidence to obtain repeatable quality releases that meet SLAs, with a metrics-driven culture. The monitoring systems typically predict likely sources of failure events so they can be improved ahead of costly failures.

Figure 5—Continuous Monitoring Maturity Level 4: Continuous Feedback

Continuous Monitoring Maturity Level 5: Continuous Improvement

Figure 6 shows key characteristics of People, Process, and Technology evident at this level of maturity. At this level there is a high degree of knowledge and confidence regarding Continuous Monitoring. Dev, QA, and Ops teams collaborate using shared metrics. End-to-end processes focus on the end customer experience and more sophisticated risk-based strategies. Typical Outcomes include an extremely high confidence and satisfaction by all stakeholders, with an innovation driven-culture. Monitoring systems are able to predict, with a high level of accuracy, where improvements are needed to avoid failures in the applications, pipelines, and infrastructure components. The monitoring systems are self-correcting, allowing for experimentation. Achievement at this level provides a platform for autonomous continuous improvement strategies in which automation and intelligence drives innovation.

"Here is the fix for the thing that will break."

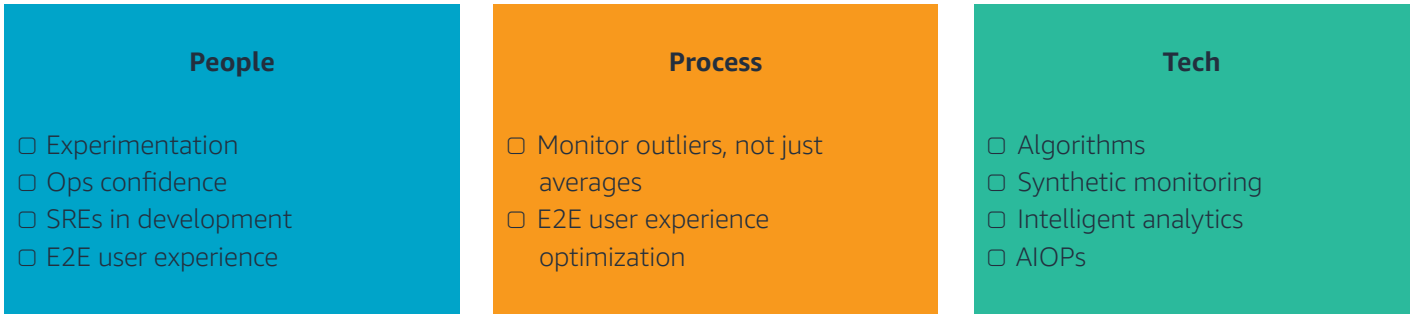

Figure 6—Continuous Monitoring Maturity Level 5: Continuous Improvement

Continuous Monitoring Maturity Assessment

While the above five levels of Continuous Monitoring maturity provide a practical guide for defining maturity against characteristics of People, Process, and Technology, they are not an absolute measure of maturity. Organizations, or specific applications within an organization, may match some of the characteristics for different levels. *Figure 7—Continuous Monitoring Maturity Model* is a useful tool to determine the "best fit" for the maturity of an organization or application within an organization. By marking the characteristics that best match, it gives a visual picture of the dominant level of maturity. This also is a quick way to determine areas to address to improve the level of maturity.

	People	Process	Tech
Chaos	▢ Silo team organization ▢ Little knowledge of monitoring ▢ Blame, finger-pointing	▢ Monitoring not part of planning ▢ Few health checks for individual components	▢ Missing tools to monitor performance of applications, pipelines and infrastructure
Continuous Integration	▢ Limited knowledge of monitoring ▢ Monitoring in backlog ▢ Ad hoc training	▢ Lack of E2E plan ▢ Repeatable build process ▢ Most commits work	▢ Version management ▢ Build monitoring ▢ Painful but repeatable releases
Continuous Flow	▢ Monitoring skills ▢ Training program ▢ Risk management ▢ Cost management	▢ E2E CI/CD pipeline, and Infra availability and incidents visible ▢ Monitoring standards ▢ Response time metrics	▢ Filtered event streams for app, infra, pipeline ▢ Deploy metrics ▢ Analysis for releases
Continuous Feedback	▢ Collaboration using shared metrics ▢ Goals: SLI/O/As ▢ Mentors and Guilds	▢ Proactive trends ▢ Performance warnings ▢ Hybrid-Cloud, Multi-Cloud	▢ APM, pipelines and infra instrumented ▢ Releases automated ▢ Value stream analytics
Continuous Improvement	▢ Experimentation ▢ Ops confidence ▢ SREs in development ▢ E2E user experience	▢ Monitor outliers, not just averages ▢ E2E user experience optimization	▢ Algorithms ▢ Synthetic monitoring ▢ Intelligent analytics ▢ AIOPs

Figure 7—Continuous Monitoring Maturity Assessment Model

A simple self-assessment tool shown in *Figure 8* indicates the maturity level for people, process, and technology, and also provides an overall score.

Level 1 – Chaos: At this level typical business outcomes are: Unpredictable, Reactive, Inefficient, Wasteful.

Level 2 – Continuous Integration (CI): At this level typical business outcomes are: Good quality of software builds, problematic quality of deployment, infrastructure inefficiency.

Level 3 – Continuous Flow: At this level typical business outcomes are: Repeatable quality releases, Some bottlenecks and inefficiencies.

Level 4 – Continuous Feedback: At this level typical business outcomes are: Repeatable quality releases, metrics driven culture.

Level 5 – Continuous Improvement: At this level typical business outcomes are: High confidence and satisfaction by all stakeholders, innovation driven-culture.

This self-assessment tool gives a very rough estimate of maturity based on a small selection of practices. To obtain an accurate and detailed assessment contact Engineering DevOps Consulting at www.engineeringdevops.com.

Continuous Monitoring Maturity Level	Selection (choose one)	People Practices	Selection (choose one)	Process Practices	Selection (choose one)	Technology Practices
Level 1 Chaos		Silo team organization, Little knowledge of monitoring, Blame, finger-pointing		Monitoring not part of planning, Few health checks for individual components		Missing tools to monitor performance of applications, pipelines and infrastructure
Level 2 Continuous Integration (CI)		Limited knowledge of monitoring, Monitoring in backlog, Ad hoc training	1	Lack of E2E plan, Repeatable build process, Most commits work	1	Version management, Build monitoring, Painful but repeatable releases
Level 3 Continuous Flow		Monitoring skills, Training program, Risk management, Cost management		Infra availability and incidents visible, Monitoring standards, Response time metrics		Filtered event streams for app, infra, pipeline, Deploy metrics, Analysis for releases
Level 4 Continuous Feedback	1	Collaboration using shared metrics, Goals: SLI/O/As, Mentors and Guilds		Proactive trends, Performance warnings, Hybrid-Cloud, Multi-Cloud		APM, pipelines and infra, instrumented, Releases automated, Value stream analytics
Level 5 Continuous Improvement		Experimentation, Ops confidence, SREs in development, E2E user experience		Monitor outliers, not just averages, E2E user experience optimization		Algorithms, Synthetic monitoring, Intelligent analytics, AIOPs
Practices Scores	4	**People Practices Scores (Calculated)**	2	**Process Practices Scores (Calculated)**	2	**Tech Practices Scores (Calculated)**

Overall Maturity Score (Calculated)	2

Figure 8—Continuous Monitoring Maturity Self-Assessment Tool

Summary

This article presents a blueprint for mature Continuous Monitoring. It explains why well-engineered, mature Continuous Monitoring is important to improve failure detection, analysis, prediction, and response—and to keep them as automated as possible. Five levels of Continuous Monitoring maturity are described. The Continuous Monitoring maturity model provides a framework to assess the maturity of an organization and to identify improvements that will lead to maturity.

aws marketplace

Implement Continuous Monitoring on AWS

To succeed in today's ultra-competitive market, organizations must migrate away from a "chaos reigns" approach to software delivery to one characterized by automation, intelligence, and maturity. Achieving this requires a new approach to leadership and culture, an end to operational silos, integration of end-to-end visibility, and a commitment to automation. A critical starting point is the adoption of a continuous monitoring solution that facilitates the AWS Well-Architected design benchmarks of operational excellence and performance efficiency.

PagerDuty is a solution that can fit seamlessly into your organization's continuous monitoring framework. By ingesting data from many logging and monitoring tools into a single intelligent event, PagerDuty can integrate with AWS services to make sense of this data. For example, through its AWS EventBridge integration, PagerDuty helps you take this information and make intelligent routing decisions. You may decide to go to Amazon Lambda for automated rollbacks, or any other failover mechanism you may design.

How AWS customers are leveraging PagerDuty to simplify their continuous monitoring

PagerDuty is a real-time digital operations management platform that combines machine learning with human intelligence to get the right information to the right people at the right time. Some of the ways that customers are leveraging PagerDuty to enhance their continuous monitoring include:

- **On call management:** PagerDuty provides alerting and scheduling so your teams are ready and empowered to take fast action.
- **Incident response:** PagerDuty can automate work across teams, execute detailed playbooks, and accelerate resolutions.
- **Event intelligence:** PagerDuty applies machine learning for full incident context, real-time triaging, and personalized recommendations.
- **Analytics:** PagerDuty brings everything together to give you a better understanding of the impact issues have on your customers, teams, and bottom line
- **Visibility:** PagerDuty offers a complete view of your data and operations, in real-time, plus stakeholder notification.

Learn more about PagerDuty >

aws marketplace

Why use AWS Marketplace?

AWS Marketplace simplifies software licensing and procurement by offering thousands of software listings from popular categories like Security, Networking, Storage, Business Intelligence, Machine Learning, Database, and DevOps. Organizations can choose from the many independent software vendors in AWS Marketplace to assemble the perfect feedback mechanism for monitoring the health of their infrastructure.

Get started with PagerDuty for free >

Learn more about AWS Marketplace DevOps solutions >

Read more from this author at engineeringdevops.com >