elastic

Observability Maturity Model

A Primer for DevOps and SRE Teams

elastic.co →

Table of Contents

For many organizations, observability is becoming a critical initiative given the dramatic increase in cloud adoption and the resulting complexity of applications. A mature observability practice relies on a comprehensive understanding of your environment, along with people, process, and technology considerations. This ebook will help you answer the necessary questions to build a mature observability practice and offer a roadmap to achieve it.

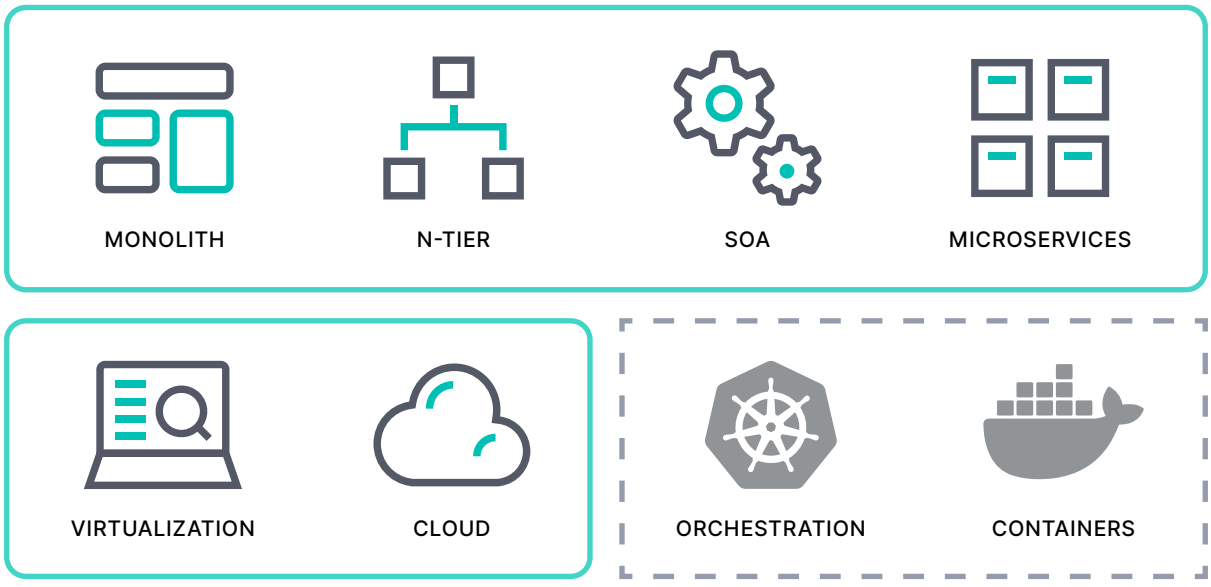
- Where do you currently stand with regards to the maturity of your observability practice?
- What does a mature practice look like?
- What is the roadmap for getting there?

Why Observability?

As modern environments continue to grow in complexity, there remains an concomitant need for ever greater visibility into the inner workings of applications and the platforms they run on.

To address this need, you'll need more data to truly understand your environment and optimize your users' experience. As such, the first tenet of observability is telemetry: gathering all the relevant logs, metrics, and traces from your infrastructure, services, and applications.

A full-featured observability solution can help you extract even more information from your monitoring data, allowing you not only to see the internal state of your systems, but also help you map dependencies, trace individual user transactions end-to-end, and uncover *unknown unknowns*: the things that may be going wrong that you didn't even know to look for.



The evolution of software architecture and cloud-native technologies

Observability is more than just data and tooling, however. A mature observability practice is built upon people, processes, *and* technology that address common pain points. These include:

- In dynamic and distributed environments, no single team or individual has the complete picture of all the dependencies
- The unprecedented breadth of cloud-native technologies (1,000+ in the CNCF landscape) creates a highly complex environment
- Communication is fragmented and critical knowledge often resides in the minds (and hard drives) of a few experts
- Telemetry data is often siloed in different tools and too much time is spent on swivel-chair investigations
- Troubleshooting often takes the form of reactive firefighting resulting in a higher mean time to resolution (MTTR)
- Applications are in a constant state of flux and prone to regressions as a consequence of fast-moving and asynchronous code pipelines

Given that it can be near impossible to find the proverbial needle in the haystack with so many moving parts and technologies, the sheer complexity of modern environments has made observability an imperative for many organizations.

The Maturity Model

Over the years, observability tooling and processes have had to keep pace with a rapidly evolving technology ecosystem: what was considered plenty of visibility yesterday, is no longer sufficient today. As such, establishing a mature observability practice is a moving target and observability requirements need to be regularly evaluated and improved upon.

Elastic's Observability Maturity Model is a tool for technical leaders and practitioners to assess where their organization is in their observability journey. Within this model, maturity is evaluated across four axes of competency: People and Culture, Process and Automation, Tools and Technology, and Business and Operational Data.

The capabilities and behaviors that characterize low versus high maturity in each of these areas are tabulated below.

People and Culture

- Highly siloed functional teams
- Limited access to observability data and dashboards
- Knowledge silos with pockets of expertise residing with one or two SMEs
- Fingerpointing is common when issues arise
- Highly convoluted approval processes
- Dev teams lack the tooling or documentation to generate observability data
- New hires struggle to contribute as training resources are out of date or unavailable

- Cross-functional teams are strongly aligned
- Frequent and transparent communications across and within functions
- Expertise, feedback, and best practices are freely shared
- Blameless culture with shared responsibility for errors or failures
- Team members are entrusted with making autonomous decisions
- Dev teams have full self-serve access to observability data, documentation, and tooling
- Focused, up-to-date, and relevant training and development resources enable new hires to ramp up quickly

Process and Automation

- Repetitive tasks soak up much of the team's time
- High occurrence of human error
- Struggle to innovate and release quickly
- Teams lack visibility into the internal state of their applications and services
- Are frequently unaware of when customer experience is degraded
- Largely reactive in the face of incidents

- CI/CD automation has reduced repetitive tasks and risk of human error
- Team members are free to focus on strategic activities
- Agile and DevOps best practices enable innovation supported by gradual changes and easy rollbacks
- Observability is built into the development process
- SLO violations are proactively tracked and simple, actionable alerts are integrated into service desk workflows
- Issues are detected and resolved quickly with fast feedback loops and early warning notifications

elastic

Tools and Technology

- Multiple, disjoint tools and dashboard
- "Swivel chair" approach to monitoring and troubleshooting
- Lack of visibility into interdependencies in complex environments
- Inability to monitor cloud-native systems
- Manually established thresholds are time-consuming and prone to false alarms
- Limited telemetry collection, often due to performance overhead and storage concerns

- Single platform for observability
- Extensive telemetry (logs, metrics, traces) is collected across infrastructure and applications
- Global customer experience is continuously tracked for real and synthetic users
- Ability to correlate data and incidents across multiple systems
- End-to-end monitoring of applications in dynamic and distributed cloud-native environments (serverless, containers, etc.)
- Service and topology maps automatically visualize interdependencies
- AI/ML is used to automate root cause analysis, detect anomalies, surface correlations, analyze trends, and be predictive
- Smart approaches to customize and tune data collection and retention to optimize storage costs and minimize impact on the systems being monitored

Low Maturity — **High Maturity**

Business and Operational Data

- Decisions are based on opinion and speculation rather than data
- Inability to map technical KPIs to business goals
- Difficulty objectively measuring improvement or success without the relevant data
- Pre-defined monitoring dashboards leave little room for ad hoc analysis
- Incidents, support tickets and customer satisfaction are frequently not tracked or reported on

- Highly data-driven with democratized access to data through curated, role-based dashboards
- Well-defined custom metrics are used to track quantifiable business objectives, such as customer retention and revenue leakage
- Metadata tagging creates a rich nomenclature for analyzing data across multiple dimensions
- Can query and visualize high cardinality data sets and perform ad hoc analysis
- High granularity historical data is available for trending and root cause analysis
- Track service desk metrics such as active alerts and number of support tickets
- Track the number and MTTR of performance-related incidents
- Track SLO and SLA compliance
- Track customer satisfaction

elastic

Accelerating Your Observability Journey

In order to evolve your observability practice, you'll need to first identify the technology, culture, and visibility gaps in your environment, and prioritize the major areas of improvement to focus on.

As you build out a plan, ask yourself the following questions:

- Where do you currently stand with regards to observability?
- Where do you need to be?
- What is the roadmap for getting there?

Elastic's Observability Maturity Assessment tool is a great resource to help you get started with this process.

Observability Maturity Levels

Expert

Focus on tying business data and objectives, refining processes, and leveraging cutting edge technologies.

Intermediate

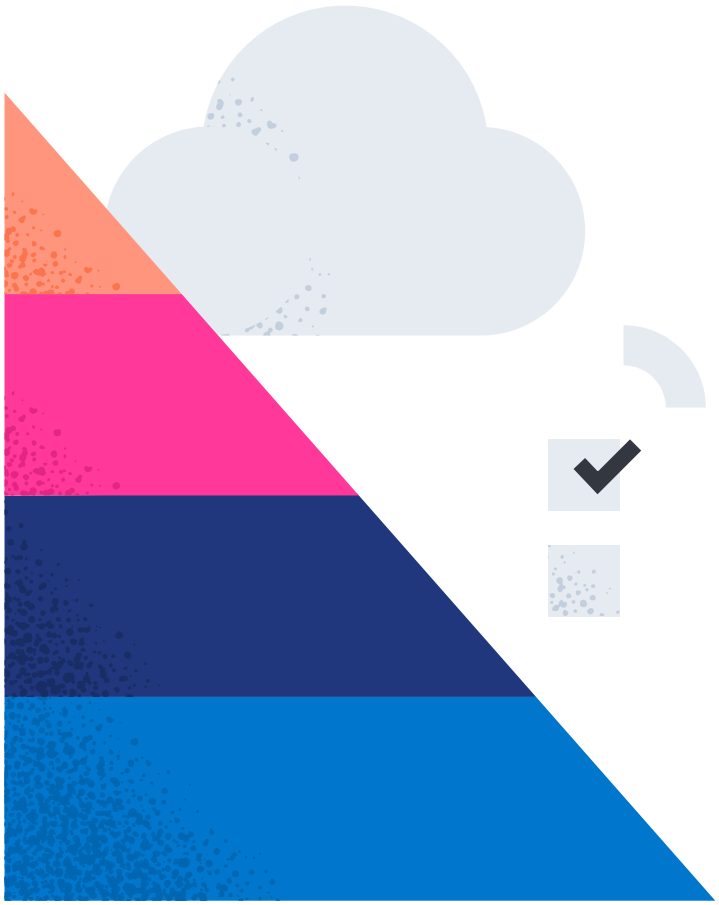
Focus on eliminating silos, democratizing data and tooling, and becoming more data-driven and customer-centric.

Beginner

Focus on addressing visibility gaps, becoming more proactive, and improving team dynamics.

Little to No Observability

Siloed teams, disjointed data sets, and reactive processes. Focus on instrumentation and automation



In the sections below, we outline a basic roadmap for maturing and evolving your culture, processes, and tooling as you map out your observability journey.

Limited or No Observability

At this level, your organization has begun to think about observability but has not yet taken the steps to implement it.

We recommend taking the following steps towards getting the people, processes, and tools in place to get the visibility you need:

- Make sure you are capturing logs, metrics, and traces for visibility into your systems and applications.
- Automate as many repetitive tasks as possible with CI/CD tools and processes in order to free up team members' time and reduce the risk of human error.
- Look into implementing modern technologies (such as cloud native, AI/ML, etc.) to help you achieve greater efficiency, scale, visibility and faster root cause analysis.
- The right set of tools can also help you gain a better understanding of end-to-end dependencies in your environment by automatically mapping dynamic infrastructure, applications and services.
- In order to detect and solve issues more proactively, you may want to consider implementing measures such as early warning notifications based on Service Level Objectives (SLOs).

Once you have the right data and analytics in place, you will be able to make more effective business and operational decisions.

Maturity Level: Beginner

At this level, your organization has started to establish an observability practice but there is still lots of room for improvement!

Although you have taken some steps towards having the people, processes, and tools in place to get the visibility you need, you can step up your game with the following recommendations:

- Identify and address any visibility gaps across your systems and applications — are you capturing all the relevant logs, metrics, and traces you need for deep and granular visibility? Is observability built into the code (for example, with actionable logs, published metrics and distributed tracing) to expose the internal state of the application?
- Create greater transparency and alignment by encouraging frequent communication and collaboration across functions.
- If troubleshooting is still largely reactive, consider implementing measures such as early warning notifications based on SLOs, integrated service desk alerting workflows and fast feedback loops to reduce MTTR.
- More extensively utilize modern technologies (cloud native, AI/ML, etc.) for greater efficiency, scale, visibility and root cause analysis. To achieve a better understanding of your environment, look into implementing tools that automatically map and trace end-to-end dependencies across dynamic infrastructure, applications and services.
- Aim to be more data-driven and customer-centric by measuring how you're doing in terms of customer experience and business objectives

With the right data and analytics in place, you will be able to more conclusively demonstrate improvement and showcase the success of your key business and operational initiatives.

Maturity Level: Intermediate

At this level, your organization is well on its way to establishing a mature observability practice!

You more or less have the people, processes, and tools in place to get the visibility you need but there are still some gaps. Implementing the following recommendations will help you achieve the highest level of maturity:

- Close any visibility gaps across your systems and applications by capturing all the relevant logs, metrics, traces, and user experience data you need for deep and granular visibility.
- Build observability and automation more effectively into the development process by ensuring that all team members have easy access to relevant data, technology, documentation and training.
- Establish a blameless culture to foster greater autonomy and trust among team members.
- Make sure you have the real-time insight you need by automatically mapping and tracing end-to-end dependencies across dynamic infrastructure, applications and services.
- Identify additional opportunities to utilize modern technologies (cloud native, AI/ML, etc.) for greater efficiency, scale, visibility and root cause analysis.
- Evaluate how you can be even more data-driven and customer-centric.

Maturity Level: Expert

At this level, your organization has achieved a high level of maturity with regards to Observability on par with the most cutting edge organizations.

Your organization likely exhibits the following best practices:

- You have almost comprehensive visibility with depth and granularity across all your systems, applications, and users.
- Team members have self-service access to data, tools, documentation and training.
- You have successfully built observability into the development process, and have automated (almost) everything with CI/CD tools and processes.
- You are effectively utilizing modern technologies (cloud native, AI/ML, etc.) for efficiency, scale, visibility and root cause analysis across most of your applications and services.
- You have an in-depth understanding of your environment, and can automatically map and trace end-to-end dependencies across dynamic infrastructure, applications and services.
- You are proactive rather than reactive. You have visibility into the digital user experience, and the data and analytics to detect and resolve issues before they impact the bottom line.
- Your organization is regularly able to communicate and collaborate seamlessly across functions, and there is likely a high degree of autonomy and trust among team members.
- Your organization is highly data-driven and customer-centric. You are able to utilize business and operational data to make effective decisions and demonstrate success.

The Business Value of Observability

Observability has immeasurable benefits to the business both in terms of revenue protection and strategic planning.

With a mature observability practice in place, organizations can::

- Decrease the number and severity of incidents
- Detect and resolve issues before they impact the bottom line
- Increase customer satisfaction and loyalty
- Make informed business and operational decisions based on data rather than speculation
- Quantify the impact of strategic initiatives

Next Steps

Ready to dive in further? Check out the following observability resources from Elastic.

- 2022 Gartner® Critical Capabilities for APM and Observability
- Leveraging Observability to Build Better Applications
- Observability Trends in 2022: A Look into the Future
- The Total Economic Impact of Elastic Observability and Security Solutions
- Elastic Named a Visionary in the 2022 Gartner® Magic Quadrant™ for APM and Observability

Search. Observe. Protect.

Elastic makes data usable in real time and at scale for enterprise search, observability, and security. Elastic solutions are built on a single free and open technology stack that can be deployed anywhere to instantly find actionable insights from any type of data — from finding documents, to monitoring infrastructure, to hunting for threats. Thousands of organizations worldwide, including Cisco, Goldman Sachs, Microsoft, The Mayo Clinic, NASA, The New York Times, Wikipedia, and Verizon, use Elastic to power mission-critical systems. Founded in 2012, Elastic is publicly traded on the NYSE under the symbol ESTC. Learn more at elastic.co.

AMERICAS HQ
800 West El Camino Real, Suite 350, Mountain View, California 94040
General +1 650 458 2620, Sales +1 650 458 2625

info@elastic.co