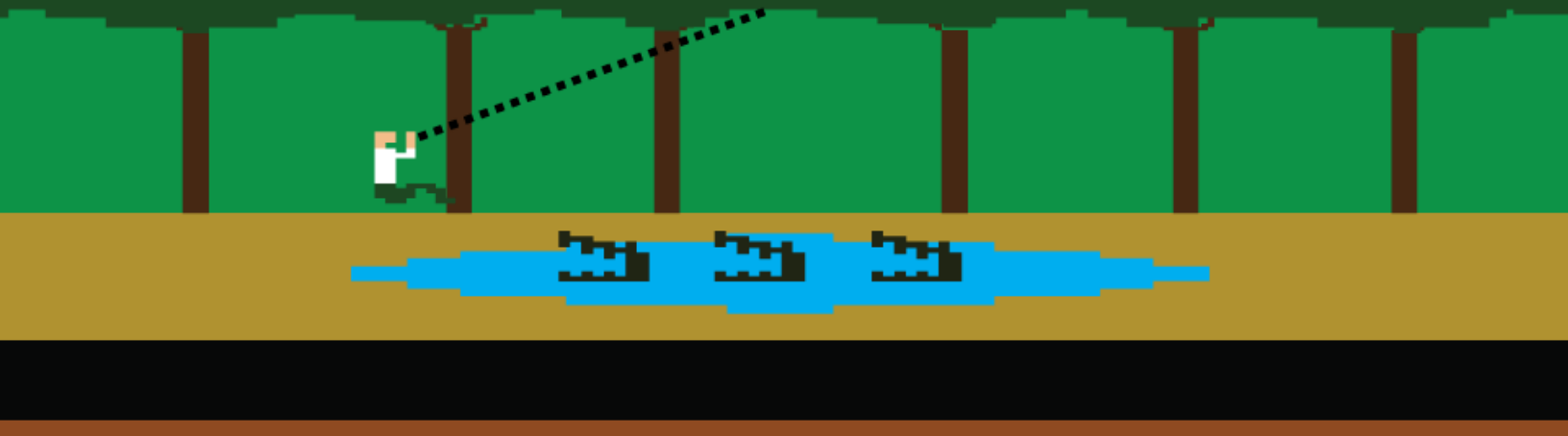


SELECTING A CLOUD MONITORING SOLUTION: PITFALLS, TRADE-OFFS, AND GOTCHAS



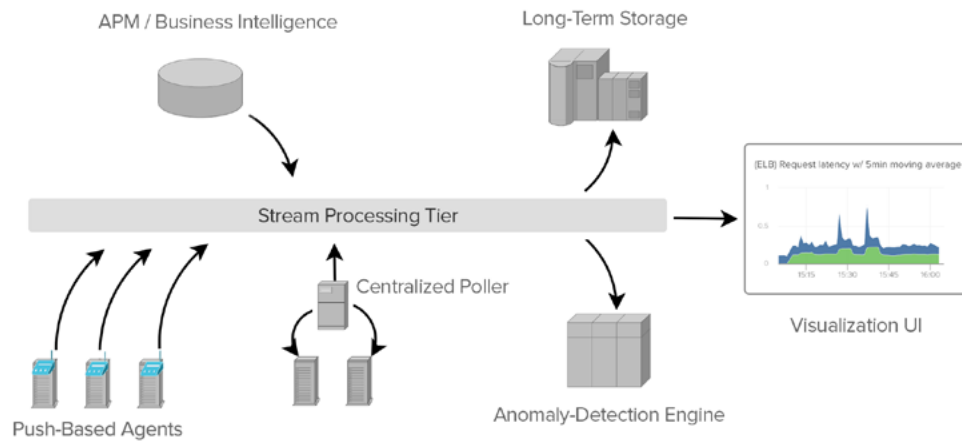
Every modern engineering team needs a cloud monitoring and metrics infrastructure. The cost of downtime alone makes monitoring operational metrics a necessity. If you don't have a monitoring strategy today, chances are that you're spending thousands of dollars and man hours on myriad ad hoc efforts put in place by your various engineering teams as they struggle to solve their individual needs.

While every IT organization needs a centralized monitoring infrastructure, building one is no easy task. The number of both commercial and open source monitoring tools has exploded in the last 10 years, and it can be very difficult to differentiate them based on their claims, all of which sound quite similar.

With that in mind, we'd like to offer a few important considerations that we hope will help you compute ROI and narrow your focus as you search for the perfect monitoring system.

Monitoring Dissected

Modern monitoring systems are normally composed of three or four components: data collectors, a stream processing tier, the persistence layer, and the analysis layer. Data collectors, as their name implies, make and transmit measurements from the entities we're interested in monitoring. Collectd and Diamond are very popular open source data collectors that were designed primarily to collect system stats (CPU, memory et.al), while DropWizard Metrics is a data collector that was designed to take



measurements inside a running Java Application. It's common for an organization to need to run a few different kinds of data collectors, both proprietary and open source, and it's a common pitfall of centralized monitoring systems that they cannot accept measurements from many different types of collectors.

The stream processing tier accepts measurements from various types of collectors and processes them in different ways. Common operations in the stream processing tier include alerting, de-duplication (getting rid of redundant measurements), aggregation and summarization (mathematically combining multiple measurements), and routing copies of each measurement to multiple upstream systems which will use the data in different ways. Heka and Riemann are both good open source stream processing systems, while Statsd is an extremely popular albeit less flexible collector/stream processing crossover tool. It is a common pitfall to omit a stream processing tier altogether; many monitoring systems implement collectors that are hard-coded to emit data directly to a custom persistence layer as a means to "lock-in" the user, and prevent them from using the collected data with other monitoring systems.

The persistence layer stores measurements to disk and makes them available to the analysis layer, which presents the data to users. There are many common pitfalls associated with these layers related to data formatting, long-term lossy storage and display.

Opensource != Free

If you don't have an existing general-purpose monitoring infrastructure, chances are good that you already have some open source monitoring tools in use by individual teams. This is not a bad thing: open source monitoring tools are, by and large, well built, secure, and dependable. It's entirely possible to build a centralized monitoring infrastructure using nothing but open source tools, but the cost of such an undertaking is often underestimated.

There is, for example, no commonly agreed upon data model for transmitting and storing monitoring data. Each open source monitoring tool is generally optimized for its intended problem domain. Hence a centralized polling tool that focuses on availability data might collect data every 5 minutes and transmit it in a textual comma-delimited format, while a process emitter might collect data every .5 seconds and emit it in a binary format.

Integrating these collectors with a persistence layer that expects yet another interval and format, along with all of the other data from all of the other monitoring efforts, is not a trivial undertaking. Proper attention must be paid to myriad technical minutiae related to integrating these tools and keeping them resilient to failure, and this complexity generally increases linearly with scale. The reality today is that properly building and running an in-house open source monitoring infrastructure requires a dedicated and skilled telemetry team.

Proprietary Tools Are No Panacea

Proprietary monitoring tools (SaaS or otherwise) can offset many of the burdensome necessities of a monitoring system composed of well integrated open source software, usually paying for themselves in the process. SaaS tools add the benefit of rapid installation, making it possible to quickly bootstrap a scalable and reliable monitoring infrastructure with a minimum of expertise.

The trade-off is the same as it has always been with proprietary software. You lose flexibility with respect to what you can monitor, how your data is stored, and/or how it may be accessed and repurposed later. A proprietary monitoring system that comes with a proprietary agent that emits to a proprietary persistence layer will collect only whatever data the vendor has specified, it will store your data for however long the vendor has specified, and it will most likely require you to employ the vendor's UI to interact with your data.

A common problem with proprietary monitoring systems occurs when they do not meet the specific needs of every engineering team in your organization, and ad hoc open source efforts crop up and negate whatever savings the proprietary system might have initially represented.

Important Questions to Ask Before Investing

When investing in a monitoring system, be it your first or a replacement of your current one, you ought to consider a few easily overlooked requirements - things that may not seem important now but will end up costing you more down the road:

- **Who is your monitoring for?** There is a relatively small subset of metrics that is equally as important to the operations engineer, the product manager and the C Suite. Even within the engineering organization, there is a spectrum of needs: from Ops engineers who need CPU metrics to developers, your people care about different real-time metrics. Investing in a separate tool for each group in your organization is costly and inefficient. A good monitoring system should be able to equip everyone with the data they need - including the business executives who unexpectedly knock on your IT shop's door for urgent reports.

Practically everything we do as a business can be measured and analyzed. Your monitoring tool should be able to provide the ability to do just that – measure and analyze everything and anything - from the get-go.

- **Will the new monitoring solution play well with the tools your team already uses and likes?** If your Ops engineers are used to, say, PagerDuty, will switching to a new alerting system wreck havoc on their productivity? Will the new monitoring system enable your developers to instrument their code and see the performance of the features they created? Investing in a system that invites everyone to plug and play is far more rewarding than creating silos in engineering work.
- **How many metrics do you really need?** It's tempting to monitor EVERYTHING, but too many metrics can be redundant and confusing, while at the same time falsely disqualifying some proprietary solutions that charge per metric. You should choose metrics carefully, and prune what you don't use.
- **What's the learning curve?** Solutions that take too long to learn, or are too difficult to use simply don't get adopted. Any engineer in the org should be able to add new metrics in minutes.
- **Do alerts and UI visualizations source from the same data set?** It's a common pitfall with multiple collectors to use one collector for alerting and another for visualization. Engineers should be able to visualize the data that alerted them.
- **Does this system lock me to a particular vendor or tool?** Consider the difficulty of replacing each of the layers in the system you're considering with something else a year from now. The system shouldn't make you re-architect everything just to replace one layer; each layer should be well integrated with, but insulated from the others.

There will undoubtedly be more questions and considerations depending on your situation – the important thing we wanted to convey is that it is worth putting time into a thorough research and comparison now, rather than pay for it later.

Once you have decided what cloud monitoring system to invest in, your ROI calculation will include the value of real-time metrics provided. For an in-depth business analysis of real-time metrics, we teamed up with David Linthicum (formerly GigaOm) to prepare a research report, "Real-Time Metrics for Improved Business Operations and Agility". **Grab your free report.**