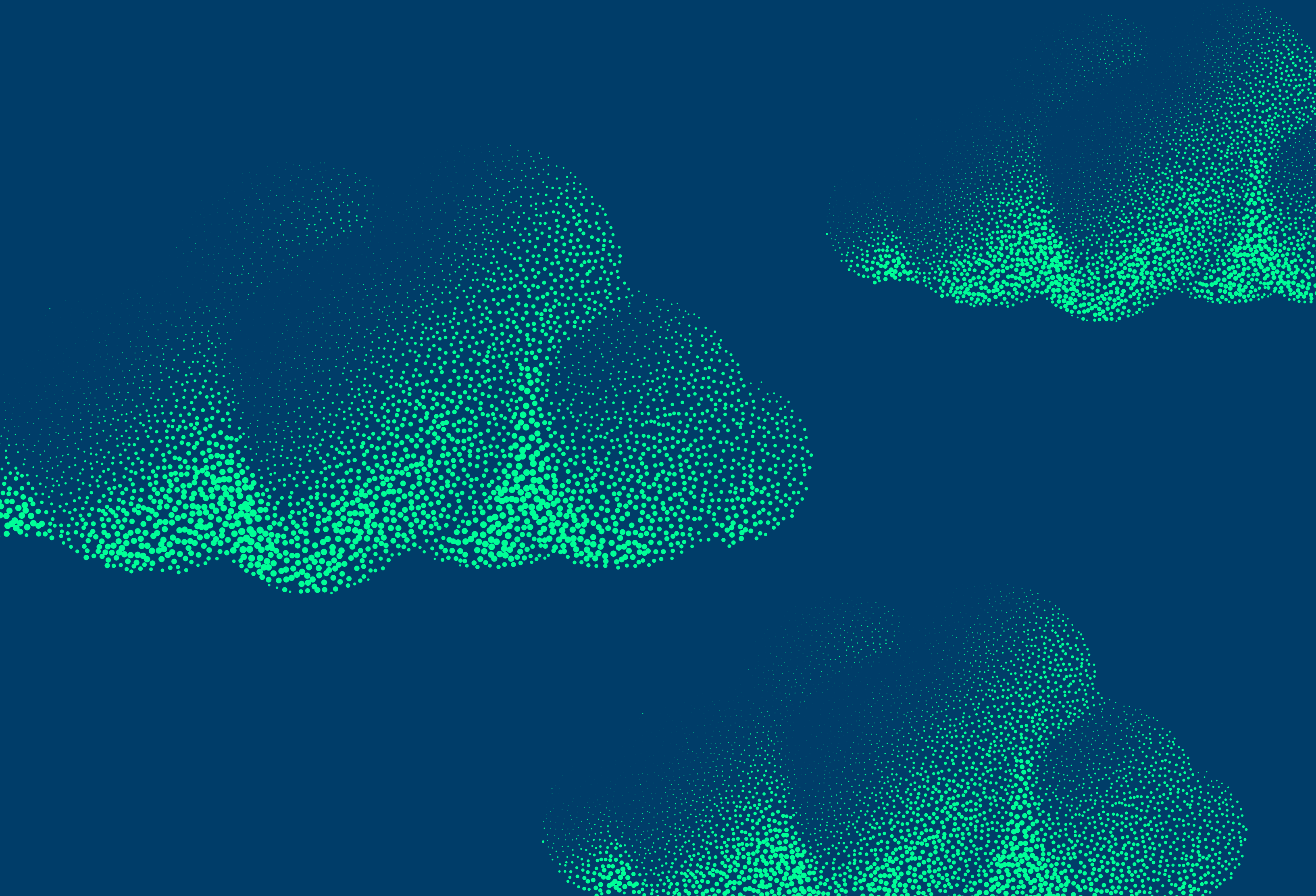
BUYER’S GUIDE

chronosphere

Prometheus-Native Monitoring SaaS Solutions: Buyer’s Guide

Ready to stop managing your own Prometheus? Here’s your buyer’s guide.



Executive Summary

The world of monitoring has fundamentally changed. Today's monitoring tools were not designed for the complex, dynamic, and interconnected nature of cloud-native architecture. Companies need a monitoring solution that is as scalable, reliable, and flexible as the cloudnative apps they need to monitor.

Prometheus' single binary implementation for ingestion, storage, and querying makes it ideal as a light-weight metrics and monitoring solution with quick time to value — perfect for cloudnative environments. But simplicity and ease has its trade-offs: as organizations inevitably scale up their infrastructure footprint and number of microservices, you need to stand up multiple Prometheus instances and start to deal with high availability and data locality issues - all of which requires significant management overhead.

Because of the time and challenges involved with running your own Prometheus deployment, many organizations are exploring moving to a hosted, or a managed metrics and monitoring SaaS offering. To ensure a smooth transition to the solution and avoid future lock-in, it's critical that the SaaS monitoring solution be fully Prometheus-native.* This whitepaper explores the key capabilities that organizations need to consider when selecting a Prometheus-native monitoring solution.

KEY QUESTIONS TO ASK YOUR VENDOR

- Is the vendor's solution Prometheus-Native? What components are run by the customer versus the vendor?
- What is the vendor's guaranteed uptime? How do they measure availability and what is their track record? What is the total cost of ownership of the solution? How do costs scale as you grow?
- What mechanisms does the vendor offer for providing visibility into data and cardinality growth and what tooling do they provide to keep growth under control? Does the vendor solution provide downsampling capabilities for historical data? Can these policies be unique by account? Can they be different for different subsets of data?
- Where does the vendor's SaaS solution run? Is it in a different cloud region to my production environment?
- How big is the vendor's largest hosted customer (unique time series over a period of time, ingested metrics/sec)?

Functional Requirements

High Availability and Reliability

Under the strain of increased data volume and cardinality, monitoring systems become unreliable: they lose data or experience downtime. Without monitoring, teams are flying blind and won't be able to respond to issues in real time. To make sure this doesn't happen to you, pay close attention to the availability and reliability of your SaaS vendor.

Service Level Agreements (SLAs)

When discussing SLAs, most people jump straight to "how many nines of uptime does the vendor offer?" This is a critical question as you'll want to have a monitoring solution that is more highly available than your production environment, meaning ideally you'll be looking for 99.9% uptime. Beyond the actual number itself, it's important to look at how the vendor defines and monitors the SLA and at what point they notify customers. A best-in-class solution will proactively monitor their own systems for downtime and count any period greater than a few minutes of the system not being accessible as downtime and immediately notify customers.

SLA Checking

Proactive SLA checking is a great start, but a simple ping check against an endpoint does not tell you much about whether the system is performing as expected — it only tells you that it is returning 200s successfully. A proper SLA check for a hosted monitoring solution should check the basic read and write paths to ensure that data is persisting as expected and no data is lost.

Dedicated Endpoint

One of the leading causes of downtime in SaaS monitoring solutions is due to noisy neighbors. This is when another tenant in a multi-tenanted SaaS environment impacts performance of other tenants. To avoid this, you would ideally want a dedicated environment with dedicated endpoints that are not shared with other customers.

WHAT DOES PROMETHEUS-NATIVE MEAN?

A Prometheus-native SaaS solution must provide:

- Prometheus ingestion protocol support
- 100% PromQL compatibility
- Prometheus Alert Manager definition support
- Prometheus Recording Rule support
- Grafana dashboard support

KEY QUESTIONS TO ASK YOUR VENDOR

- What is the vendor's uptime guarantee?
- How does the vendor define downtime? What does the vendor use to monitor for downtime?
- In the event of vendor downtime, how quickly are you notified?

Cloud Provider Choice/Circular Dependency Protection
Most engineers are familiar with the concept of a circular dependency — when A depends on B, but B also depends on A — but most don't think about this in the context of their SaaS monitoring service and production. If your production applications are hosted in the same cloud provider and region as your SaaS monitoring solution, you've got a circular dependency. The primary risk is that if there is a service disruption in that region, you could have a simultaneous outage of both your production environment and your monitoring solution. This is one of the worst possible times to incur an outage to your monitoring solution as it is the system that is meant to inform you of your production outage. Best practice is to host your SaaS solution in a different region, AND with a different IaaS cloud provider if possible, to avoid this circular dependency.

KEY QUESTIONS TO ASK YOUR VENDOR

- What region and cloud is the SaaS offering hosted in? Can I request a different region or cloud?
- Does the vendor offer single tenancy and multi-tenancy SaaS offerings?
- If multi-tenant, how does the vendor protect against "noisy neighbors"?

Security and Administration

You'll need to understand what fine-grained security and access controls the vendor has put in place to protect your data. You'll want to ensure that all user or machine communication is encrypted and that both user and service accounts leverage secure authentication systems.

Administration and Access Control
Administration and access control is also a vital piece of the security puzzle. It's critical for administrators to get access to make changes to metadata like dashboards and alerts etc. However, if users have these same permissions, the results could be catastrophic. The administrators must be able to restrict users to read various subsets of the underlying data (metric data itself). For example, developers must be able to see other teams' environments, but should not be able to make potentially breaking changes.

Single Tenancy
As mentioned above, when your SaaS solution uses a single tenant infrastructure, it provides a great reliability and availability advantage to customers. It also provides a significant security advantage: all of your data is completely isolated from other customers using the SaaS service. This makes the possibility of cross-account contamination or other types of breaches, extremely low.

Complete End-to-end Managed Solution
Even though they are SaaS offerings, several of the solutions on the market are not complete end-to-end offerings. For example, in some cases, you are still responsible for running your own instance of Grafana for dashboarding and visualization and Prometheus Alert Manager for alerts. Other solutions also force you to continue running Prometheus collection instances in your own environment. This additional management overhead can ultimately prove to be very time consuming and expensive for organizations as well as lead to other significant challenges like single points of failure and lower availability/reliability. That's why in many cases it makes more sense to work with a SaaS solution that is Prometheus-native, instead of pure managed Prometheus (i.e., has full compatibility with Prometheus as listed earlier, but doesn't have some of the limitations that come with pure managed Prometheus). That will eliminate the need for additional tooling you run yourself.

Cost & Control

Cloud-native environments emit a massive amount of monitoring data — especially as developers add more labels to their metrics causing massive cardinality spikes. As monitoring data volumes grow, so do costs and this growth is often outpacing the growth of overall infrastructure costs.

To combat data growth (and ultimately costs) organizations must make decisions based on business value about what data is kept, for how long, and at what resolution. This not only helps keep cost under control, but can make it easier and faster to find the data needed to solve problems.

KEY QUESTIONS TO ASK YOUR VENDOR

- How is user access management and role-based access control handled? Can users be granted granular permissions and access based on their role?
- Does the vendor integrate with SSO/SAML authentication tools to automatically provision/deprovision users?
- How are service accounts authenticated?
- What infrastructure-level security standards are in place?
- Is the vendor certified to any compliance standards, such as SOC2 Type II?
- Is the vendor offering a completely managed solution? If not, what components is the customer expected to run?

Pricing Model

First, you'll want to evaluate the vendor's pricing model. Unfortunately, there is no single standard for pricing Prometheus-native monitoring solutions, so it can be hard to compare apples to apples on price. The key things to focus on are determining if the pricing is fair and easy to understand, and if it's predictable and only grows as your value from the tool grows.

Downsampling and Aggregation

One of the ways many organizations deal with increasing volumes of data is by downsampling and aggregation. Downsampling allows for historical metric data to be read with better performance while reducing storage costs over time. Aggregation of metric data focuses on better query performance of high cardinality metric data. Many tools on the market today rely on native Prometheus methods of downsampling and aggregation called Recording rules, which work by ingesting all the raw metric data into the time series database (TSDB) first before reading them and then producing the aggregated and downsampled metric data back into the TSDB.

One of the many disadvantages to this approach is that it does not reduce the overall volume of data stored since all raw unaggregated data must be kept in order to create aggregated data — in fact, the overall volume of data increases. Some SaaS vendors give you the ability to optionally discard the raw underlying data and only keep the downsampled or aggregated data which will heavily reduce the volume of data persisted and correspondingly reduce costs.

Retention and Resolution

There are so many different use cases for monitoring in modern cloud-native environments and each have their own set of requirements. For example, a user may want to view the per container metrics and break it down by the pod UUID as it's critical for deployments. This data will be extremely high cardinality and useful in real-time, but becomes less useful

KEY QUESTIONS TO ASK YOUR VENDOR

- How predictable is the vendor's pricing model? Is there a possibility of getting an overage?
- Is there an additional per-query charge on top of ingest and storage fees?
- What aggregation and downsampling levers does the vendor offer? At what point in the data stream does this occur — at the client side, server side, or after being written to the TSDB?

over longer periods of time — after a year, no user will care about the specifics of a particular pod since it most likely will not have existed for a very long time. Alternatively, specific aggregated metrics are useful for trend analysis over longer periods of time. Since there are so many different use cases for monitoring data, it does not make sense to try and fit a single data retention and resolution policy across all of them.

For maximum efficiency, you want to be able to pick and choose different retention and resolution policies for different subsets of your monitoring data - one that is tailored for each use case.

Visibility & Control

Being able to gain visibility into how much data each team or user is putting into the monitoring system as well as how much data is fetched is key to controlling rising costs over time. Ideally, this visibility is broken down into logical units that make sense for each company — whether that be by service, cluster, environment, or team.

Visibility goes hand-in-hand with enforcement when it comes to control. Ideally you can take the new visibility you’ve gained and use it to limit data writes or reads by group. The goal of putting these limitations in place is to ensure that one user cannot impact another’s experience.

Finally, you’ll want a solution that allows you to give the rate-limited end users an ability to resolve the issue without requiring the end user to redeploy or configure their instrumentation leads to a faster resolution. The solution does not have to lead to data being dropped — downsampling, aggregation and control over resolution and retention are all mechanisms that allow the end user to make choices on their data in order to optimize for both the use case and cost.

KEY QUESTIONS TO ASK YOUR VENDOR

- What capabilities does the vendor provide to help with data growth and cost control? Does the vendor offer rate limiting? Can it be applied by team or label?
- Does the vendor you offer the ability to dynamically adjust the resolution of any subset of metric data?
- Does the solution provide a way to show which teams are generating the greatest volume of metrics and cardinality? What kind of cost-attribution metrics are available and can they create granular alerts and customized dashboards?

Performance and Scale

It's critical that you have a monitoring solution that you trust can scale reliably as you transition to cloud-native and continue to achieve high levels of performance. Under the strain of increased data volume and cardinality and scaling workloads, some monitoring systems struggle to ingest all the data and make it available in real time. Without real time monitoring, teams might not find out about issues right away, which can lead to prolonged customer impact.

Proven to Scale for the Future

There are two key elements of scale to consider, the first is more forward looking and will help you determine if you will outgrow the SaaS solution in either the short or the long-term. Many organizations have found that as they scale their cloud-native environment, their SaaS monitoring vendor can't keep up. To be sure you are picking a solution that meets your scale needs today and in the future, you'll want to ask the vendor to provide examples of other customers of a similar size and growth rate, and ideally speak with these customers as well.

Speed of Alerting

When something goes wrong, you want to know as soon as possible so you can begin to remediate the issue. In order to know something has gone wrong, you need to get an alert from your monitoring system. Not all SaaS vendors are created equal in this regard, so you'll want to ask how frequently alerts are checked for, with the goal of a solution that notify users of an issue within a few seconds.

Alert Generation and Management

To get up and running faster, you'll want to look for automated alert set-up. Since most services or hosts produce the same metric data, automated or templatized alerting can help engineers know about problems without a complicated set-up process. Some tooling also offers the ability to group or correlate alerts so engineers don't get overwhelmed by alert storms and can quickly get to the information they need, to start remediating the issue.

KEY QUESTIONS TO ASK YOUR VENDOR

- How many writes per second does the vendor's largest SaaS monitoring customer consume?
- How many active timeseries does the vendor's largest customer have?
- At what frequency are alerts checked? How fast are alerts generated?
- Does the vendor offer any type of alert grouping or correlation?
- How many concurrent alerts does the vendor support at their largest customer?

Support and dependability

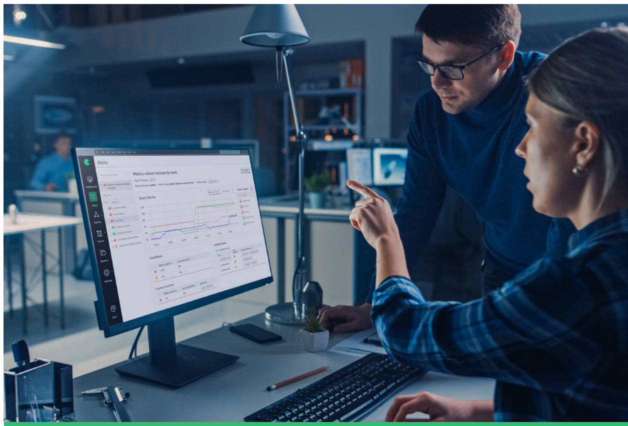
While the functional requirements are the core focus area for buyers, services, support, and pricing cannot be overlooked. This area can often be a game changer, as many organizations require strict SLAs or need a dedicated team to help them ensure success. Look for a vendor who is a trusted partner — who can bring expertise to the table and has the skills to deal with the unexpected. What services beyond the product does the vendor provide: On-boarding of historical data? Assistance setting up alerts and dashboards? Enablement sessions with users and administrators? You'll want a vendor that does all of these things.

SLA Track Record

Actions speak louder than words, it is more important to know what your SaaS vendor's historical performance on SLAs is more important than what they are promising you for the future. Remember too, that SLA definitions vary company to company, so make sure you understand the full context of how they define an outage when you make side by side comparisons. You'll want to ask not only for overall customer SLA performance, but also for the SLA performance of their biggest (and likely most demanding) customers.

Customer References

As with any vendor selection process, getting customer references is a key part of making a decision. Talking live to existing customers is critical, rather than relying on customer stories. If you have any hesitations about the vendor, talking to the customer reference can be a good way to either solidify, or dismiss these concerns.



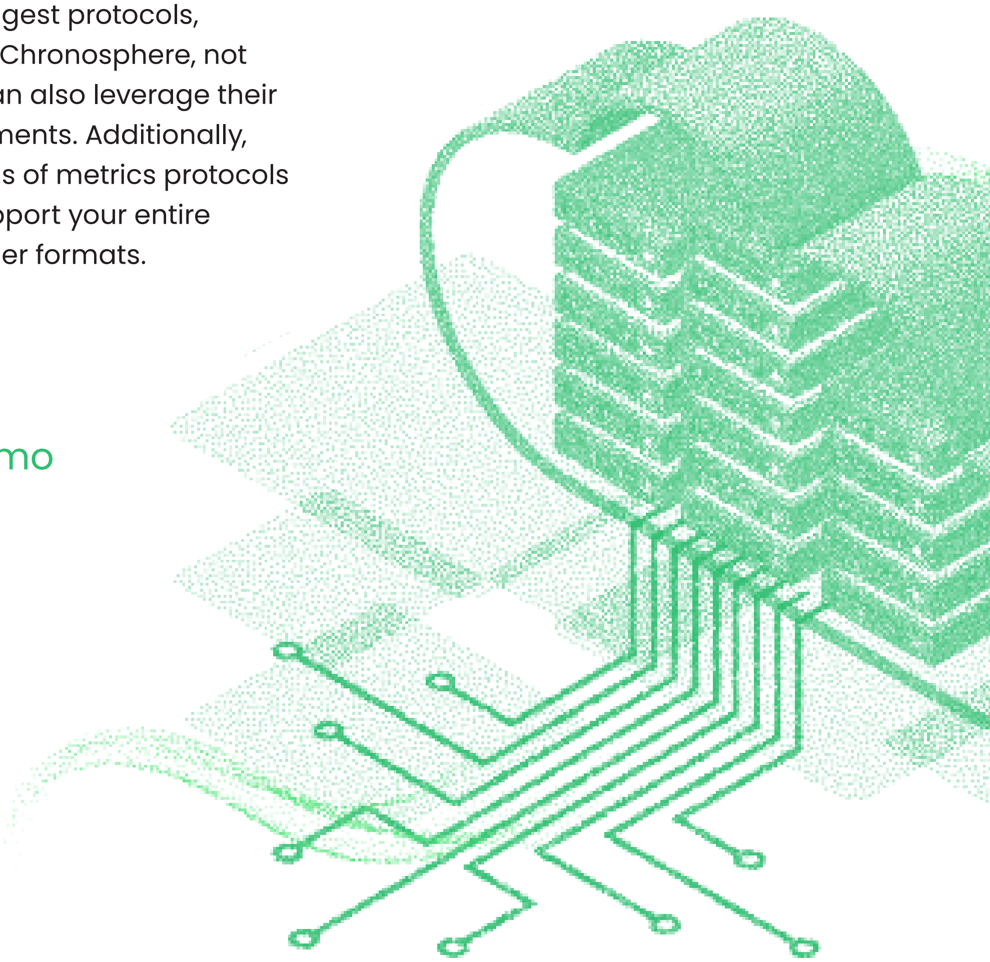
KEY QUESTIONS TO ASK YOUR VENDOR

- What is the vendor's actual delivered SLA over the past 12 months for all customers? What is the vendor's actual delivered SLA for the top 10% of customers (by size)?
- Does the vendor offer on-boarding of historical metric data?
- What on-boarding services are available? Assistance setting up alerts and dashboards? Enablement sessions with users and administrators?

Next steps: the bake-off

After you've completed a paper evaluation, the next step is to do a bake-off: Most vendors offer a free or paid pilot where you can test the capabilities and make sure it will meet your needs. Make sure you go into the pilot with a clear set of success criteria and a plan for how you will put the product through its paces.

As you go through this process, you may come to the conclusion that a pure managed Prometheus offering doesn't actually meet your criteria — instead what you need is a Prometheus-native SaaS solution. Chronosphere is the only SaaS monitoring solution built for cloud-native, providing deep insights into every layer of your stack — from the infrastructure to the applications to the business. Chronosphere is open-source compliant, and fully supports Prometheus metrics ingest protocols, dashboards, and query languages. With Chronosphere, not only can teams avoid lock-in, but they can also leverage their existing Prometheus and Grafana investments. Additionally, Chronosphere supports older generations of metrics protocols (Graphite, StatsD, etc), meaning it will support your entire environment, even as you migrate off older formats.



Learn more and request a demo at **chronosphere.io**.