



ИНДЕКСЛОГ

Индекслог\_Поиск. Индекслог\_Аналитика.  
Индекслог\_Безопасность.

# Введение в ARM

<https://rr-th.com/>

# Оглавление

<u>Зачем нужен мониторинг производительности приложений (APM)?</u>	<u>3</u>
<u>Приложения обеспечивают работу компаний</u>	<u>3</u>
<u>Низкая производительность равна потерянному доходу</u>	<u>4</u>
<u>Современные облачные приложения очень сложные</u>	<u>4</u>
<u>Что именно представляет из себя APM?</u>	<u>6</u>
<u>Видимость</u>	<u>6</u>
<u>Устранение задержек во времени ожидания и ошибок</u>	<u>7</u>
<u>APM за рамками производства</u>	<u>7</u>
<u>Как работает APM?</u>	<u>8</u>
<u>Агенты APM</u>	<u>8</u>
<u>Инструменты и конфигурация</u>	<u>9</u>
<u>Анализ</u>	<u>9</u>
<u>Ключевые термины</u>	<u>10</u>
<u>Распределенная трассировка</u>	<u>10</u>
<u>Spans</u>	<u>11</u>
<u>Транзакции</u>	<u>11</u>
<u>Трассировка приложений</u>	<u>12</u>
<u>Мониторинг реальных пользователей</u>	<u>12</u>
<u>Выбор инструмента APM</u>	<u>13</u>
<u>Технические возможности</u>	<u>13</u>
<u>Простота использования</u>	<u>14</u>
<u>Варианты развертывания</u>	<u>14</u>
<u>Поддержка открытых стандартов / открытых данных</u>	<u>14</u>
<u>Архитектура и масштабируемость</u>	<u>15</u>
<u>Безопасность</u>	<u>15</u>
<u>Возможности за пределами APM</u>	<u>16</u>



# Зачем нужен мониторинг производительности приложений (APM)?

*Небольшое примечание: в этом руководстве мы будем рассматривать [мониторинг производительности приложений \(APM\)](#) как одну из основ [мониторинга](#). Вместе с журналами и метриками, APM играет важную роль в построении мониторинга систем.*

## Приложения обеспечивают работу компаний

Приложения - это публичное лицо современных организаций. С их помощью мы взаимодействуем с продуктами и услугами, будь то витрина интернет-магазина, приложение для совместной поездки или различные инструменты для совместной работы и повышения производительности, которыми мы пользуемся ежедневно. Когда мы добавляем пару наушников в свою корзину или составляем идеальный плейлист, мы взаимодействуем с приложениями и формируем свое мнение о них. Восприятие бренда, лояльность и рейтинги основаны на этом опыте.



# Низкая производительность равна потерянным доходам

Подумайте о приложениях, которые вы используете на работе или дома. Что приходит на ум? Насколько они просты в использовании? Как быстро реагируют? Как часто ошибаются? Влияют ли ответы на эти вопросы на ваше восприятие компаний, создающих эти приложения?

Когда приложения работают не так, как ожидалось, это оказывает прямое воздействие на бизнес. На карту поставлены доходы и доверие клиентов.

Если функция совершить покупку не работает, пользователи уйдут в другой магазин, что приведет к потере продаж. Если время отклика слишком медленное или часто возникают ошибки, даже лояльные пользователи могут решить полностью отказаться от использования продукта или услуги.

Каждая организация знает, что означает время простоя для ее итоговой прибыли.

Среднее время обнаружения (MTTD) и среднее время реагирования (MTTR) имеют свою цену.



## Современные "облачные" приложения слишком сложные.

В связи с тем, как быстро меняются способы разработки и поставки приложений, а также количеством сложностей, которые возникают в результате этих изменений, необходимость в комплексном мониторинге производительности ощущается все более остро.

Внедрение архитектур микросервисов эффективно изменило способ создания и поддержки приложений, предлагая непрерывную поставку и масштабирование, что ранее было недоступно.

Однако распределенная и полиглотная (написанная с использованием различных языков/фреймворков) природа этих архитектур добавляет сложности.

Принципы непрерывной поставки и автоматизации позволяют часто обновлять код, но при этом способность отслеживать влияние на производительность становится более важной, чтобы можно было быстро устранить проблемы или вернуть изменения, если развертывание идет не так, как планировалось.

Принципы облачно-нативной разработки позволяют командам постоянно обеспечивать пользователям лучший опыт использования приложений, быстро реагировать на отзывы и при необходимости вносить изменения. Пользователи не видят, что все эти процессы происходят в фоновом режиме, обеспечивая работу каждого их запроса. Как же команды следят за этими приложениями?

Как люди, мы не можем осмысленно обрабатывать такой объем данных. Нам нужно организованное представление с четким путем к пониманию. На помощь приходит APM.

# Что именно представляет из себя APM?

APM - это решение для сбора, мониторинга и анализа сквозной производительности и поведения транзакций через ваши приложения (и сервисы, от которых они зависят). С момента начала взаимодействия пользователя с приложением и до достижения им желаемых результатов APM отслеживает его опыт.

Благодаря прямой связи с пользовательским опытом APM помогает связать IT с бизнес-целями. Взаимосвязь может быть не совсем 1:1, но она имеет положительную корреляцию. Обеспечивая постоянное понимание того, как работают ваши приложения, APM позволяет действовать проактивно, а не реагировать на проблемы после получения жалоб клиентов через систему тикетов или даже социальные сети.

## Видимость

После того как приложение проинструментировано, полученные данные позволяют точно определить, что происходит внутри приложения. APM отслеживает транзакции на всем их пути, поэтому каждый запрос и ответ регистрируются и измеряются, независимо от сложности архитектуры.

Связывая все эти traces вместе, APM дает вам полную видимость производительности вашего приложения - от общей картины сервисов (и их взаимодействия) вплоть до информации на уровне кода.



*У вас может возникнуть вопрос – а разве нет журналов, которые рассказывают нам о том, что происходит внутри наших приложений? Журналы действительно дают нам ценные данные с контекстной информацией о типах ошибок и других происходящих событиях. Но журналы не показывают сквозной путь пользователя, и именно здесь трассировка APM (и распределенная трассировка) заполняет пробелы. Сопряжение трассировок с журналами и метриками обеспечит вам единую видимость вашей экосистемы.*

## Устранение задержек во времени ожидания и ошибок

APM отслеживает два атрибута, связанных с пользовательским опытом: продолжительность транзакций и ошибки. Сколько времени обычно занимают HTTP-запросы? Сколько ошибок обычно мы видим в течение часа?

Проблемы с задержкой возникают, когда ответ на запрос занимает больше времени, чем обычно. В качестве примера можно привести ситуацию, когда пользователь нажимает на кнопку, чтобы посмотреть товар, а загрузка ответа занимает более нескольких секунд.

Ошибки означают, что произошел непредвиденный результат. Это происходит, когда запрос не был успешно завершен.

Конечной целью любого расследования является устранение проблем. Чтобы сделать это эффективно, команды проводят анализ первопричины, чтобы точно определить, какой компонент вызывает проблему. Можем ли мы быстро определить, какой сервис является узким местом в производительности? Можем ли мы точно определить вызов функции или метод, ответственный за эту проблему? С APM ответ на оба вопроса - "да".

## APM за рамками производства

Мы часто говорим об APM в контексте работы пользователей с существующими приложениями, но APM вступает в игру раньше, оказывая влияние на работу различных команд и сред. Например, APM предоставляет разработчикам обратную связь, необходимую для создания кода, что приводит к ускорению цикла "разработка-тестирование-развертывание".

Компании используют APM на всех этапах, чтобы помочь сотрудникам более эффективно создавать, проверять, внедрять и контролировать приложения. Если организации используют APM в значительной части отделов, они могут улучшить и создать более эффективные рабочие процессы, что позволит больше времени уделять инновациям (и меньше времени тратить на решение проблем).

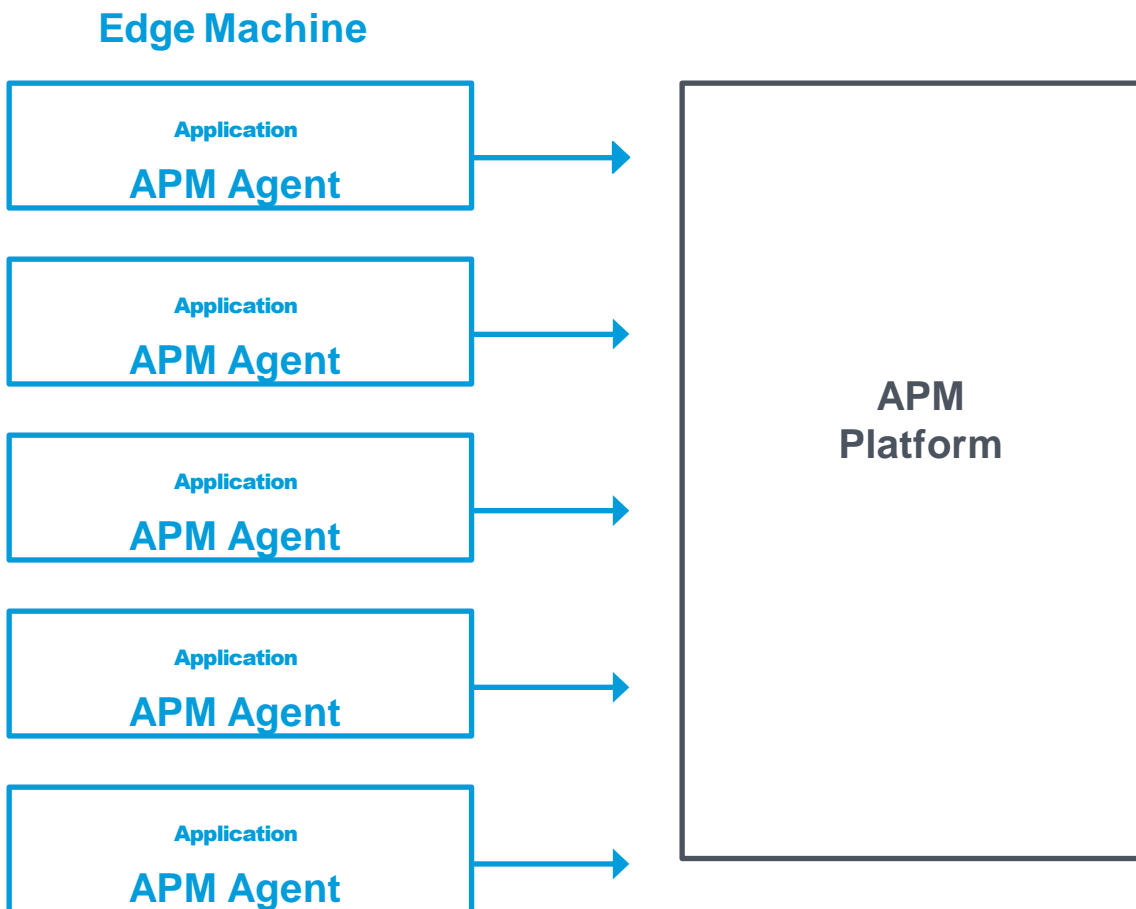
Мы увидели преимущества; теперь давайте погрузимся в то, как работает APM.

# Как работает APM

APM работает путем инструментации каждого аспекта вашей системы, который вы хотели бы контролировать (хосты, контейнеры, приложения и т.д.), а затем отправляет собранные данные о производительности в выбранное вами место для анализа.

## APM агенты

Агент APM - это библиотека, плагин или расширение, которое отслеживает показатели производительности, описанные выше. В зависимости от того, что вам нужно отслеживать (и на каком языке это написано), вам может понадобиться один или несколько агентов. После определения всего, что вы хотите контролировать, вы разворачиваете агенты APM для каждой из этих частей. Хотя агенты отличаются по производителям, большинство агентов инструментуют ваш код, собирают данные о производительности, а затем отправляют их на сервер или коллектор.





# Инструментирование и конфигурация

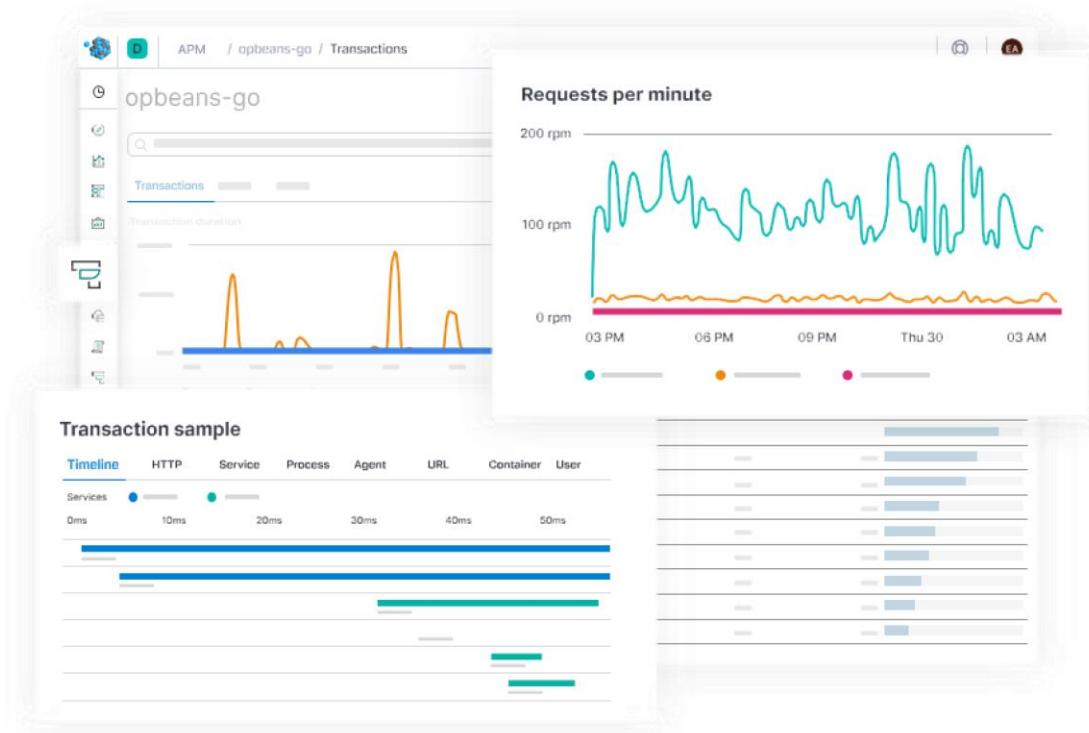
Инструментирование - это процесс расширения кода вашего приложения для получения данных трассировки. После установки необходимого агента(ов), вы можете инструментировать определенные части вашего приложения, которые вы хотите проанализировать, чтобы собрать и отправить данные о производительности в выбранное вами место, обычно конечная точка определяется вашим APM инструментом.

Вы, вероятно, захотите настроить такие вещи, как имена окружения, частоту выборки, инструменты и метрики, чтобы помочь вашим командам легко идентифицировать и анализировать данные, которые поступают в ваш инструмент APM. Обычно это делается с помощью пользовательского интерфейса и API инструмента или непосредственно в переменных среды.

## Анализ

После отправки данных о производительности в выбранное вами место, вы готовы к анализу данных. Большинство инструментов имеют пользовательский интерфейс, который поможет вам определить ошибки, проблемы задержки и другие аномалии, которые влияют на ваших пользователей.

Это отличное место для начала расследования сообщений о проблемах, а еще лучше - для выявления проблем до того, как они затронут ваших пользователей. Точно определите, какие службы (вплоть до уровня кода) испытывают проблемы, чтобы ускорить анализ первопричин и MTTR.





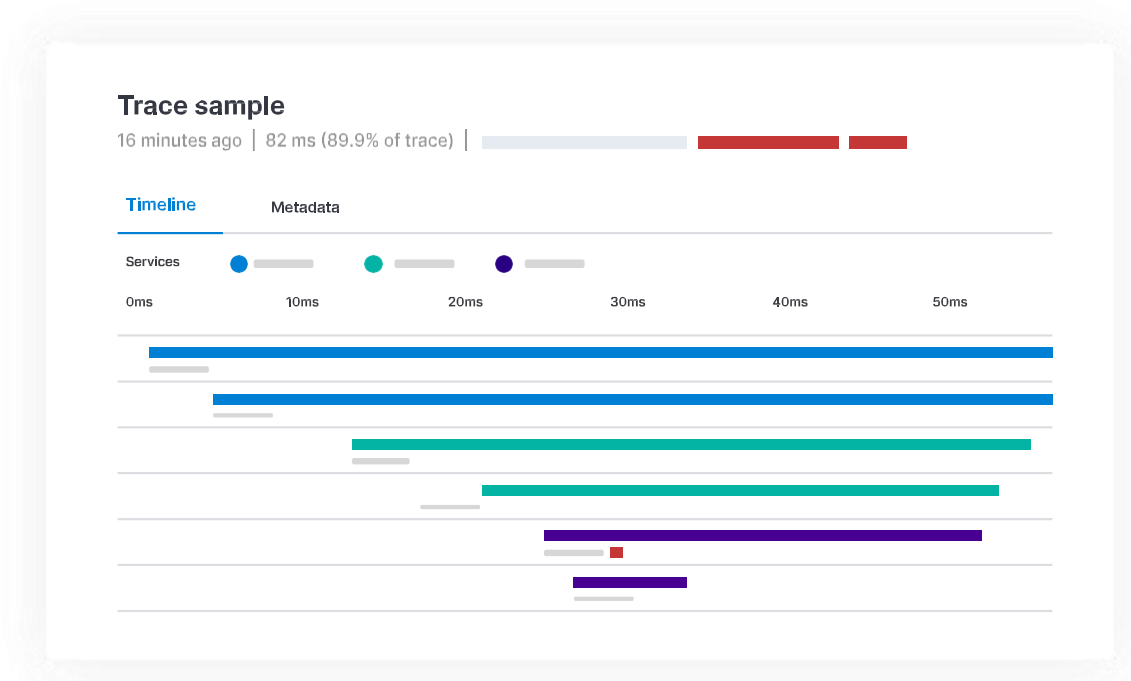
# Ключевые термины

У каждого инструмента APM могут быть немного разные определения для этих терминов, но в целом они собирают одну и ту же информацию, чтобы обеспечить пользователю представление о производительности его приложений.

## Распределенная трассировка

Отслеживая все запросы, начиная с начального веб-запроса к вашей внешней службе и заканчивая запросами к вашим внутренним службам, распределенная трассировка позволяет вам анализировать производительность всей архитектуры микросервисов в одном представлении.

Распределенная трассировка позволяет легко обнаружить узкие места, отображая полные события по службам, а затем по каждому запросу в рамках этой службы. Выявление ошибок и других проблем на уровне кода позволяет ускорить расследование и MTTR.



# Spans

Каждая единица, или часть рабочего процесса, называется span. Spans - это то, что вы обычно видите в каскадном представлении (модель «Водопад») инструмента анализа APM, обычно изображаемые в виде горизонтальных полос. Эти spans являются основой распределенной трассировки. Spans измеряются от начала до конца деятельности и содержат информацию о выполнении определенного пути кода.

Общие признаки span включают:

- Время начала
- Время окончания
- Имя
- Тип

## Транзакции

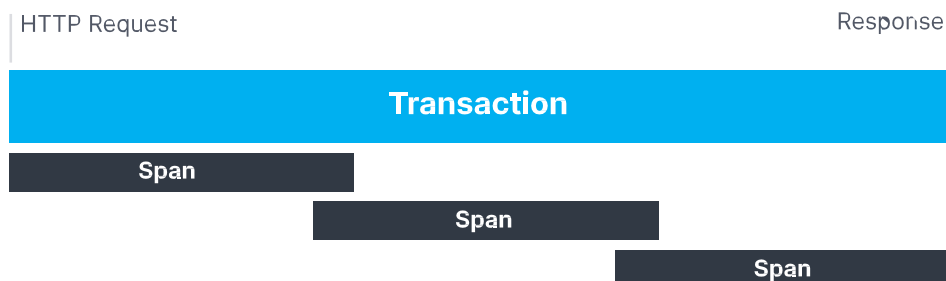
Транзакции - это тип span, который описывает событие и может также включать несколько spans.

Несколько примеров сделок включают:

- Запрос к вашему серверу
- Пакетное задание
- Фоновая работа

Транзакции имеют дополнительные атрибуты, связанные с ними, например, данные о среде, в которой регистрируется событие:

- Сервис: среда, фреймворк, язык и т.д.
- Хост: архитектура, имя хоста, IP и т.д.
- Процесс: args, PID, PPID и т.д.
- URL: полный, домен, порт, запрос и т.д.
- Пользователь: (если указано) электронная почта, ID, имя пользователя и т.д.

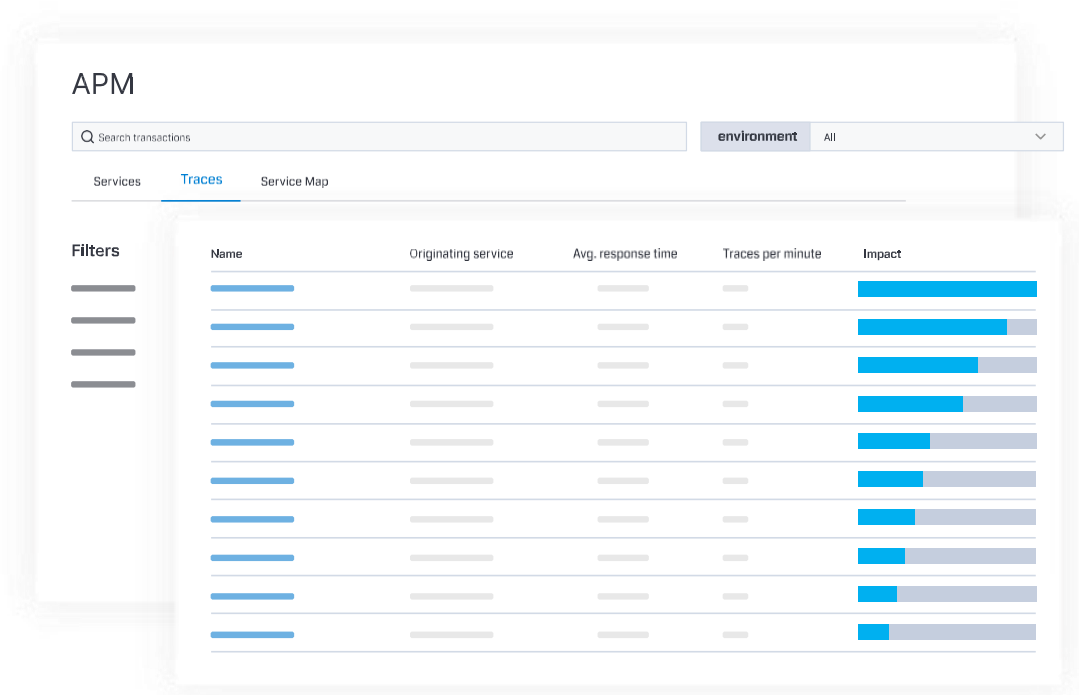


# Трассировка приложений

Трассировка нужна для измерения состояния и продолжительности запроса, сделанного через ваше приложение. Spans и транзакции вместе образуют trace, соединяя точки от запроса пользователя до ответа.

Вы увидите отдельные trace для таких действий, как обработка платежа, обработка завершенного заказа и обновление статуса доставки.

В сочетании с журналами и метриками приложения (и других аспектов вашей инфраструктуры), trace обеспечивают полную видимость всей вашей экосистемы.



# Мониторинг реального пользователя

Мониторинг реальных пользователей (RUM) фиксирует прямое взаимодействие пользователей с клиентами в производственных средах. Например, RUM показывает, как именно пользователь взаимодействовал с вашим сайтом, а также время отклика и возникшие ошибки. Это выполняется агентом в браузере пользователя, который собирает данные о производительности и отправляет их в инструмент APM для анализа.

# Выбор инструмента АРМ

Как и в любом процессе выбора инструмента, начните с четкого определения ваших требований. При составлении списка требований учитывайте рост и будущие потребности. Запланируйте проверку концепции (РОС), чтобы убедиться, что инструмент действительно соответствует вашим потребностям. Работают ли возможности, демонстрируемые поставщиком, так, как ожидается, в вашей среде? РОС помогает избежать неожиданностей и скрытых расходов на более поздних этапах процесса.

## Технические возможности

Составьте контрольный список необходимых функций с учетом ваших конкретных технических потребностей. Будьте внимательны и детализированы в своих требованиях. Хотя список не является исчерпывающим, здесь приведены некоторые области, о которых следует подумать:

## Простота использования

В процессе работы над техническими требованиями к функциям оцените простоту использования и доступность этих функций. Подумайте о разных пользователях, которые будут взаимодействовать с инструментом, и учтите их опыт, ожидания и рабочие процессы при ответе на вопрос о простоте использования. Расширьте свою оценку за пределы опыта начала работы и перейдите к оценке постоянных эксплуатационных расходов.

- Как быстро вы можете создать инструмент для приложения и пройти путь от нуля до инсайта?
- Является ли пользовательский интерфейс интуитивно понятным и простым в использовании?
- Сколько усилий потребуется для развертывания решения?
- Насколько легко обновить платформу? Как насчет агентов?
- Можно ли централизованно управлять агентами из одного места?
- Какие потребуются усилия для увеличения (или уменьшения) масштаба?

## Развертывание варианты

- Важно, чтобы выбранный инструмент APM поддерживал ваши предпочтения в использовании программного обеспечения.
- Предпочитаете ли вы вариант SaaS для снижения операционных и административных расходов?
- Нужна ли вам опция, которая поможет сократить ежегодные расходы на выбранного вами поставщика облачных услуг (AWS, Azure, Google Cloud и т.д.)?
- Вам нужно самоуправляемое предложение, потому что облако не подходит по причине стоимости или соответствия нормативным требованиям?
- У вас многооблачная или гибридная стратегия, и вы хотите использовать решение APM приближенное к вашему объему нагрузки, чтобы сократить расходы на передачу данных или задержки?

## Поддержка открытых стандартов / открытых данных

В области мониторинга начинают разрабатываться и приниматься открытые стандарты, такие как OpenTelemetry (образованный путем объединения двух других открытых стандартов, OpenTracing и OpenCensus) в качестве стандартной, нейтральной к производителям инструментальной базы. Цель этих инициатив - помочь разработчикам легко перенести свое приложение на другое APM-решение без необходимости повторного инструментирования.

Если у вас есть существующие приложения, инструментарий которых использует открытый стандарт, например Jaeger, переход на инструмент с его поддержкой может ускорить и упростить миграцию. Открытые стандарты также помогают защитить ваши инвестиции на будущее.

# Архитектура и масштабируемость

Оцените, есть ли у выбранного вами **инструмента APM** надежная основа и архитектура для работы с текущими и будущими масштабами. Ваша способность быстро и эффективно анализировать и решать проблемы работы приложений зависит от производительности инструмента APM. Обязательно проведите нагрузочное тестирование инструмента APM в процессе оценки (или просмотрите эталонные данные, если это невозможно), чтобы убедиться, что он может обрабатывать ожидаемый объем входящих данных и запросов в вашей среде, не проседая под нагрузкой.

- Построен ли он она на простой архитектуре? Или представляет собой лоскутный фундамент, который рано или поздно даст трещину?
- Поддерживает ли он высокую доступность?
- Можно ли его легко масштабировать для обработки всплесков объема данных мониторинга?
- Существуют ли какие-либо ограничения на объем (приложений, метрик, запросов и т.д.), который он может обрабатывать?

# Безопасность

Процесс анализа безопасности должен быть основной частью вашего процесса оценки инструмента. Обязательно учитывайте следующие два аспекта:

1. Обязательства поставщика APM по обеспечению безопасности при создании и поставке инструмента
  - Нужны ли агентам APM, развернутым в ваших приложениях, особые условия?
  - Зашифрован ли и защищен ли трафик между внутренними компонентами должным образом?
  - Имеет ли инструмент необходимые сертификаты (особенно в SaaS)?
  - Использует ли инструмент сторонние расширения? Безопасны ли они?
2. Возможность защиты и контроля доступа к инструменту APM
  - Интегрируется ли инструмент с вашей корпоративной системой управления доступом к идентификационным данным?
  - Поддерживает ли инструмент управление доступом на основе ролей с гранулярной моделью разрешений?

## Возможности за пределами APM

Потребность в консолидации инструментов и единого мониторинга стимулирует многие организации объединить APM с журналами и метриками - двумя другими "столпами мониторинга" - чтобы помочь упростить анализ и сократить MTTR. Оцените, поддерживает ли APM единое видение, отображая журналы и метрики в контексте данных трассировки приложения. Как насчет ввода данных за пределами типичного IT-ландшафта, таких как настройка пользователей или запросы в службу поддержки клиентов?

