

ОБОСНОВАНИЕ ПИЛОТА · ДЛЯ РУКОВОДИТЕЛЯ

Ваши инженеры должны устранять проблемы, а не собирать контекст по десяти системам.

Предлагаю протестировать AutoDevOps — инструмент для автоматического расследования инфраструктурных инцидентов на наших Linux-серверах. Он подключается к существующему мониторингу, собирает технический контекст и формирует проверяемый отчёт с доказательствами. Исправления автоматически не применяет. Предлагаю провести ограниченный бесплатный пилот и измерить время до полезного контекста, точность выводов, затраты на LLM и трудозатраты на внедрение. Обработка LLM выполняется через OpenRouter с передачей данных расследования после маскирования секретов. По результатам примем решение, стоит ли масштабировать решение и приобретать лицензию.

Меньше повторяющейся работы

AutoDevOps берёт на себя часть начального сбора и сопоставления данных: состояние Linux-хостов, метрики, журналы и сведения о базах данных.

Знания становятся доступнее команде

Структурированные доказательства и предлагаемые проверки помогают инженерам продолжать расследование без полной зависимости от одного специалиста, который знает всю инфраструктуру.

Контролируемый риск

Проверки выполняются read-only инструментами; модель не получает произвольный shell и не применяет исправления в продакшене. Решение и изменяющие действия остаются за инженером.

Не обещаем абстрактные **-70% MTTR**. Предлагаем измерить влияние на ваших инцидентах в пилоте.

Что оцениваем в пилоте

- время до первого полезного, подтверждённого контекста / гипотезы;
- долю корректных и проверяемых выводов, причины ошибок и ложные выводы;
- количество ручных проверок / переключений между инструментами, которые удалось исключить;
- затраты LLM и реальные ограничения обработки данных;
- усилия на интеграцию, поддержку и реакцию инженера.

Безопасность и данные

- Только чтение: агент ничего не меняет на хостах, нет shell — фиксированный каталог проверок.
- Ни одного входящего порта на хосте: агент сам подключается к мастеру по mTLS.
- Базы — под read-only учёткой: ClickHouse с readonly=1, PostgreSQL с pg_monitor.
- Секреты маскируются на входе мастера; креды — в OpenVault; каждый вызов инструмента и модели — в журнал аудита.
- Роли и доступ по группам хостов, вход через Keycloak (OIDC), дневной лимит расходов на модель.

Мастер разворачивается у вас. Для работы выбранной языковой модели часть данных расследования передаётся через OpenRouter после маскирования секретов.

Как проходит пилот

1. Master на VM клиента; установка через Ansible / Docker Compose.
2. Агенты — deb-пакет из репозитория мастера.
3. Prometheus / Alertmanager, выбранные мессенджер и LLM подключаются в UI.
4. Первые расследования — по поступившим сигналам.
5. Владелец продукта помогает на каждом шаге.

Пилот бесплатно. Расходы на LLM оплачиваются отдельно и ограничиваются дневным лимитом.

Совместимость

- Хосты: Ubuntu 20.04, 22.04, 24.04 · Debian 12 · железо и VM, в том числе за NAT.
- Метрики и алерты: Prometheus или VictoriaMetrics · Alertmanager · node_exporter.
- Базы: ClickHouse (каждый узел кластера) · PostgreSQL 16 и 18.
- Мессенджеры: Slack · Telegram · Zulip.
- Kubernetes — в разработке.

Контрольные прогоны

18 из 18 расследований — первопричина названа верно; ≈ \$0,24 — средняя стоимость модели на расследование (Claude Sonnet 5.5), не цена AutoDevOps; 0 ложных фактов в названной причине. Шесть типовых аварий × 3 прогона, слепая оценка по эталону сценария. Это контролируемый набор, а не гарантия точности на проде.