🤖 Создание первого Managed Agent с Claude

🚀 Создание первого Managed Agent с Claude

Ключевые тезисы:

- Claude Managed Agents — это самый быстрый способ создания production-ready агентов.
- Основная цель — абстрагировать сложности (хостинг, масштабирование, безопасность) и позволить разработчикам сосредоточиться на задачах и логике агента.
- Архитектура разделяет «мозг» (агент-луп) и «руки» (исполнение инструментов) для повышения безопасности, надежности и производительности.
- Агенты работают на основе событий (events), а не простых запросов-ответов, что обеспечивает надежность и наблюдаемость.

🎯 Ключевые концепции

Эволюция платформы Claude

1. **Messages API (2023):** Сырой доступ к модели. Разработчики сами реализовывали все примитивы (управление контекстом, агент-луп и т.д.).
2. **Agent SDK:** Появился *harness* (обвязка) для вызова Claude Code. Разработчики всё ещё сами управляли хостингом и масштабированием.
3. **Claude Managed Agents:** Первый *managed harness*, который берет на себя масштабирование и production-готовые компоненты (песочница, observability, runtime для инструментов).

Три основных ресурса Managed Agents

- **Агент (Agent Endpoint):** *«Мозг»*. Определяет персону и возможности: модель, системный промпт, инструменты (MCP-серверы, навыки).

- **Окружение (Environment):** *«Руки»*. Контейнер или пространство, где агент выполняет действия (например, Anthropic Cloud или ваш собственный контейнер).
- **Сессия (Session):** Связывает агента и окружение, позволяет стримить события пользователю.

🔧 Практический воркшоп: Создание SRE-агента

Цель: Создать агента для автоматического реагирования на инциденты (Incident Response).

Шаг 1: Определение агента

Задаем модель (Claude Opus 4.7), системный промпт (роль SRE, задача — дебаг инцидентов) и список доступных инструментов (метрики, логи, деплои).

Шаг 2: Определение окружения

Настраиваем, где будет работать агент. Можно использовать managed-инфраструктуру Anthropic или **принести свой контейнер (BYOC)**. Указываем сетевые правила (белый список URL).

Шаг 3: Предоставление данных (контекста)

Загружаем файлы с логами и метриками через Files API. **Контекстный инжиниринг** — ключевая часть, где разработчик тратит основное время, решая, какие данные дать агенту.

Шаг 4: Создание сессии

Связываем ID агента, ID окружения и загруженные ресурсы. Сессия — это точка входа для взаимодействия.

Шаг 5: Стриминг событий и реализация локальных инструментов

- Агент общается **событиями** (сообщения пользователя, вызовы инструментов, ответы агента), а не просто токенами. Это улучшает UX и observability.
- **Локальные инструменты** (например, `get_metrics`, `get_recent_deploys`) реализуются на стороне разработчика. Агент-луп работает в облаке Anthropic, а исполнение инструментов — локально или в вашей инфраструктуре.

🎭 Результат

Агент анализирует логи, проверяет метрики и последние деплои, находит root-cause инцидента (например, исчерпание пула соединений БД из-за конкретного коммита) и предлагает действия по исправлению.

💡 Преимущества архитектуры Managed Agents

- **Разделение «мозга» и «рук»:** Повышает безопасность (изоляция credentials), позволяет независимо масштабировать компоненты.
- **Снижение задержки (Latency):** Разделение привело к сокращению **TTFT (Time To First Token) на >90%**.
- **Сохранение состояния (Session Persistence):** Сессии и их состояние (idle, running, terminated) сохраняются в облаке. При перезагрузке приложения или потере соединения сессию можно возобновить.
- **Production-ready из коробки:** Управление надежностью, кэшированием, компрессией контекста, observability.

🚀 Возможности для расширения

- **Навыки (Skills) и Под-агенты (Subagents):** Для параллельного выполнения задач и управления контекстом.

- **Память и Dreaming:** Агент может **самостоятельно анализировать свои логи** (Dreaming), чтобы запоминать предпочтения пользователей и исправления, становясь самообучающимся.
- **Результаты (Outcomes):** Можно задавать критерии успешного выполнения задачи, а не только список действий для агента.
- **Хранилища (Vaults):** Безопасное управление учетными данными на основе разделенной архитектуры.
- **Вебхуки (Webhooks):** Запуск агентов или изменение состояния сессий по внешним событиям.
- **Консоль разработчика:** Готовый дашборд для observability и управления агентами.

Выводы: Claude Managed Agents позволяет быстро создавать мощных, безопасных и готовых к продакшену агентов, абстрагируя инфраструктурные сложности. Ключевые преимущества — event-ориентированная архитектура, разделение логики и исполнения, а также встроенные продвинутые функции (память, outcomes, vaults).
