Пользовательские приложения в MikroTik

30 мая 2026 года

Пользовательские приложения (custom apps) в MikroTik — это приложения, которые не входят в стандартный каталог MikroTik, но могут быть добавлены администратором вручную. Они позволяют расширить функционал RouterOS за счёт сторонних решений — например, кастомных веб-сервисов, скриптов автоматизации или специализированных сетевых утилит.

Как это работает
В RouterOS есть меню `/app` — это каталог приложений, которые можно развернуть в несколько кликов. Каждое приложение может состоять из:
* одного или нескольких предварительно настроенных контейнеров;
* необходимой конфигурации RouterOS (правила брандмауэра, трансляция адресов и т.д.) — она применяется автоматически.

Стандартный каталог приложений поддерживается MikroTik: образы контейнеров берутся из публичных реестров (Docker Hub, GCR, Quay), а параметры конфигурации можно редактировать перед включением приложения.

Пользовательские приложения дают возможность добавить в этот каталог **свои** решения — например:
* собственный веб-интерфейс для мониторинга;
* скрипт автоматизации сетевых задач;
* контейнер с нестандартным сервисом (например, для IoT или аналитики трафика);
* модифицированную версию официального приложения с индивидуальной настройкой.

Ключевые особенности пользовательских приложений
1. Источник образов: в отличие от стандартных приложений, пользовательские могут брать образы контейнеров из приватных реестров или локальных файлов.
2. YAML-конфигурация: администратор задаёт параметры развёртывания в YAML-файле. В нём можно указать:
* имя приложения (`name`);
* описание (`descr`);
* URL веб-интерфейса (`ui-url`);
* настройки сети (IP-адрес, интерфейс);
* лимиты памяти (`memory-max`);
* монтирования томов;
* переменные окружения;
* правила брандмауэра и т.д.
3. Автоматизация: при запуске пользовательского приложения RouterOS автоматически:
* создаёт виртуальный интерфейс (VETH);
* настраивает мост и маршрутизацию;
* применяет правила брандмауэра для проброса портов;
* запускает контейнер с заданными параметрами.
4. Управление жизненным циклом: пользовательские приложения можно:
* запускать/останавливать;
* перезапускать;
* обновлять (с принудительной очисткой старых данных);
* удалять.
5. Мониторинг: в меню `/app print` отображаются:
* статус (запущено/остановлено);
* потребление памяти (`memory-current`);
* размер образа (`app-size`);
* размер данных (`data-size`);
* категория;
* описание;
* URL веб-интерфейса (`ui-url`).

Требования к среде
Чтобы использовать пользовательские приложения, нужно:
* установить пакет `container`;
* включить режим контейнеров (может потребовать перезагрузки устройства);
* обеспечить достаточный объём ОЗУ для работы контейнеров;
* использовать внешнее хранилище (USB/SATA/NVMe) для оптимальной производительности;
* выбрать диск с минимальной скоростью чтения/записи 100 МБ/с и 10 K IOPS (можно проверить командой `/disk/test`).

Практические примеры
Пользовательские приложения полезны для:
* развёртывания кастомного веб-интерфейса для управления сетью;
* запуска скрипта мониторинга с графиками и алертами;

* интеграции со сторонними сервисами (например, облачным резервным копированием);
* тестирования новых сетевых утилит до их включения в официальный каталог;
* создания специализированных шлюзов (например, для IoT-протоколов);
* настройки индивидуального прокси или кэширующего сервера.

Пошаговый процесс развёртывания
1. Подготовить YAML-файл с описанием приложения (имя, образ контейнера, настройки сети, монтирования и т.д.).
2. Указать URL пользовательского хранилища приложений (`app-store-urls`) в настройках `/app`.
3. Выбрать приложение из каталога (в т.ч. пользовательского) через GUI (кнопка **Setup**) или CLI (`/app setup`).
4. Выбрать диск для установки (система автоматически обнаружит доступные отформатированные диски).
5. Дождаться автоматической загрузки и извлечения образа контейнера.
6. Дождаться настройки сети (создание VETH-интерфейса и моста).
7. Дождаться создания правил брандмауэра для доступа к веб-интерфейсу.
8. Получить URL веб-интерфейса (`ui-url`) и начать работу с приложением.

Важные нюансы
* Безопасность: тщательно проверяйте образы контейнеров и YAML-конфигурации перед развёртыванием — они могут содержать уязвимости или вредоносный код.
* Совместимость: убедитесь, что архитектура устройства (arm64, x86) и версия RouterOS поддерживают нужное приложение.
* Ресурсы: отслеживайте потребление памяти и дискового пространства, особенно на устройствах с ограниченным ОЗУ или SPI-flash.
* Обновления: для пользовательских приложений нет автоматического обновления — администратор должен вручную обновлять YAML и образы.