Иммутабельные ОС: эволюция, принципы и баланс с реальностью

Балашов Кирилл Алексеевич, ООО «РЕД СОФТ»
РЭУ им. Г.В. Плеханова. Адрес: Москва, ул. Большая Серпуховская, д. 11, корпус 9, этажи 7 и 8, "Точка-кипения РЭУ".
kirill.balashov@red-soft.ru

Откуда возникла идея

Иммутабельная, или же неизменяемая операционная система – это ОС, в которой корневая файловая система монтируется только для чтения, а любые изменения либо сохраняются во временной файловой системе до перезагрузки, либо применяются путем атомарной замены образа целиком через A/B-схемы разметки диска. Такой подход стал ответом на фундаментальные проблемы классических дистрибутивов, особенно остро проявившиеся в эпоху контейнеризации.

Осознание того, что сервер или устройство должны выполнять строго ограниченный набор функций, привело к созданию операционных систем, избавленных от всего лишнего, где целевое состояние задаётся не императивными скриптами, а декларативной моделью, а агенты системы непрерывно приводят реальное состояние к желаемому.

Путь развития

Первыми шагами стали минималистичные серверные ОС, основанные на монолитном образе, с фоновым агентом обновлений, загружавшим новые образы на неиспользуемую часть диска и философией «всё – контейнер». Позже появились специализированные платформы, которые развили эти принципы: отказ от традиционного пакетного менеджера в пользу монолитных образов, запуск всех компонентов исключительно в изолированной среде и отсутствие интерактивного доступа.

Эволюция шла по нескольким направлениям. Во-первых, произошёл переход от традиционных пакетных менеджеров к механизмам вроде OSTree, которые управляют контентом через деревья фиксаций на основе жёстких ссылок в хранилище, позволяя монтировать текущее развёртывание в режиме read-only и атомарно переключаться на новые версии с сохранением перезаписываемого слоя /var и /etc.

Во-вторых, внедрялось централизованное декларативное управление через API: конфигурация узла, например, Ignition или cloud-config, задаётся в виде JSON/YAML-документа, подписывается цифровой подписью и применяется однократно при первой загрузке, после чего локальный агент непрерывно сверяет реальное состояние с желаемым, устраняя расхождения. Наконец, происходило максимальное сокращение кодовой базы хоста – за счёт статической сборки компонентов, проверки целостности цифровыми подписями и отказа от SSH.

Таким образом, от просто «урезанных» систем индустрия пришла к двум сосуществующим моделям: полностью иммутабельнной архитектуре, построенной на монолитном образе с отказом от пакетного менеджера и интерактивного доступа, и гибридным ОС, основывающихся на read-only файловой системе, атомарных обновлений и контейнерной изоляции в экосистему классических дистрибутивов.

Общие концепции

Полностью иммутабельная ОС — это специализированная система, спроектированная для работы под управлением Kubernetes: все компоненты, включая kube-proxy, CNI-плагины, системные агенты и containerd, поставляются в виде docker-образов и управляются статическими Pod'ами, SSH отсутствует на уровне ядра, а образ системы пересобирается целиком с помощью CI/CD пайплайна при любом изменении конфигурации. Такой подход даёт минимальную поверхность атаки, полную идентичность узлов и быструю замену, но требует пересборки образа при любом изменении.

Гибридная модель, напротив, берёт базовый образ ОС, монтируется через overlayfs, где нижний слой – неизменяемый, а верхний слой – перезаписываемый и сбрасываемый при перезагрузке. Сервисы выносятся в контейнеры, а управление конфигурацией и пакетами осуществляется не через пакетный менеджер напрямую, а, например, через rpm-ostree, которые создают новый «снимок» файловой системы и применяют его после перезагрузки. Прямой SSH-доступ обычно отключается, заменяясь на инструменты вроде cockpit в контейнере. Это сохраняет совместимость с оборудованием и удобство отладки ценой чуть более сложного контроля целостности.

Именно эти различия определили сферы применения. Полностью иммутабельные ОС чаще используются в больших kubernetes-кластерах, где важны безопасность, скорость масштабирования и идентичность нод. Гибридные решения, напротив, преобладают там, где необходима совместимость с нестандартным оборудованием, запуск приложений, хранящих состояние или запуск на одиночных серверах. Кроме того, гибридная модель активно применяется там, где требуется детерминированное поведение при сохранении возможности гибкой настройки.

Баланс

Выбор между иммутабельной и гибридной архитектурой – это не поиск идеального решения, а подбор подходящей архитектурной модели под конкретную задачу. Обе модели решают одну и туже проблему – отказ от децентрализованного управления в пользу воспроизводимой инфраструктуры. Иммутабельная операционная система изолирует узел полностью, превращая его в атомарный «блок». Гибридная же модель сохраняет привычное разделение на ОС и приложение, но жёстко фиксирует неизменный базовый слой, на который уже декларативно накладывается вся остальная инфраструктура.

РЕД ОС 8 Core

РЕД ОС 8 Core является гибридной иммутабельной ОС, в которой корневая файловая система смонтирована только для чтения, а перезаписываемые разделы /var и шаблонизированный /etc обеспечивают сохранение состояния приложений, логов, образов контейнеров и данных СУБД. Система ориентирована на атомарные обновления, повторяемые узлы Kubernetes и возможность запускать необходимые сервисы вне контейнеров благодаря наличию слоя с пакетами.

Одна из ключевых особенностей РЕД ОС 8 Core – это атомарное обновление на базе rpm-ostree. В отличие от полностью монолитных образов, оно действует более смешано: неизменяемые слои образа атомарно обновляются из реестра OCI, а локальный слой приложений может попакетно дообновляться из репозиториев без пересборки всей ОС. Целевое состояние описывается в Ignition-файле, обновления загружаются как новый снимок OSTree, проверяется его цифровая подпись, и после перезагрузки система атомарно переключается на новую версию, при этом все данные в /var и изменения в /etc полностью сохраняются. Это позволяет хранить состояние приложений, логи, кэш образов контейнеров, базы данных и пользовательские данные между обновлениями, а также тонко настраивать параметры системы без необходимости пересборки всего образа.

Таким образом, РЕД ОС 8 Core реализует прагматичный подход: отказ от традиционного пакетного менеджера в пользу атомарных образов сочетается с возможностью гибридной установки пакетов в локальный слой и сохранением управляемого слоя конфигурации, что позволяет легко интегрироваться в существующие производственные процессы.