



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

KVM + libvirt: production-установка гипервизора

Наш стандарт: bridge вместо NAT, LVM-thin, live migration,
бэкап по dirty bitmaps

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

К нам приходят с парком из 4-8 разнородных физических серверов: 1С с SQL, файловая шара, контроллер домена, телефония, почтовый шлюз — каждый на своём железе 2014-2018 годов, без запаса ресурсов и внятного восстановления. Консолидируем это на два KVM-хоста на Ubuntu Server LTS с libvirt, общим storage и живой миграцией: обслуживание железа перестаёт быть простым.

Почему это важно бизнесу

- Отказ одного старого сервера = стоп отдела; на KVM это переезд VM на второй хост без остановки сервисов
- Лицензии гипервизора после Broadcom стали статьёй расходов; KVM входит в ядро Linux и не требует подписки
- Плановое обновление прошивок и дисков делаем в рабочее время: VM уезжают на соседний хост
- Новый сервис поднимается за час из cloud-образа, а не за две недели закупки железа
- Один инженер обслуживает 30-40 VM через virsh и Ansible вместо обхода стойки



Ключевые параметры реализации

10.0.0

libvirt в Ubuntu 24.04 LTS при QEMU 8.2.2 — наша базовая связка для новых хостов

пакетная база Ubuntu 24.04

7.6.0+

libvirt с полной поддержкой инкрементального бэкапа: нужен QEMU 6.1+ и qcow2

libvirt, incremental backup

49152-49215

TCP-диапазон живой миграции (migration_port_min/max, qemu.conf) — только в migration-VLAN

libvirt qemu.conf

16514/TCP

порт libvirtd/virtproxysd поверх TLS; plain 16509 у нас закрыт всегда

libvirt drvrremote

3%

порог CPU steal на хосте, выше которого мы разносим VM по хостам

наш стандарт

1 GiB

размер hugepages под VM с СУБД и сервером 1C, balloon у таких VM отключён

наш стандарт



Базовая раскатка production-хоста с нуля

Что настраиваем

Типовой сценарий: 1-2 сервера 2U, 2 сокета, 256-384 ГБ ECC, NVMe в RAID-10, 2x10G в bond LACP, офис 40-75 рабочих мест

Как мы это делаем

- 1 BIOS: включаем VT-x/VT-d (или AMD-V/AMD-Vi), отключаем C-states глубже C1, профиль питания Performance; проверяем `egrep -c '(vmx|svm)' /proc/cpuinfo` и `kvm-ok`
- 2 Ubuntu Server 24.04 LTS (libvirt 10.0, QEMU 8.2) + `qemu-kvm`, `libvirt-daemon-system`, `virtinst`, `ovmf`, `cloud-image-utils`
- 3 Сеть: `netplan` поднимает `bond0` → `br0` с адресом хоста, `stp: false`; в `libvirt` заводим сеть `forward mode='bridge'`. NAT-сеть `default` удаляем (`net-destroy/net-undefine`)
- 4 Storage: LVM thin-pool на NVMe (тома отдаём VM блочными устройствами), отдельный `dir`-пул под `qcow2` и `iso`-пул; в XML дисков `cache='none'`, `io='native'`, `discard='unmap'`
- 5 Профиль VM: `virtio-blk/virtio-scsi`, `virtio-net` с `vhost` и `multiqueue`, `qemu-guest-agent` через канал `org.qemu.guest_agent.0`, UEFI из OVMF, `cpu mode='host-model'`

РЕЗУЛЬТАТ

Хост уходит в эксплуатацию за один рабочий день вместе с приёмочными тестами: пересборка VM из `cloud-init`, перезагрузка хоста с автостартом доменов (`virsh autostart`), проверка гостевого агента и заморозки ФС. Параметры зафиксированы в Ansible-роли — второй хост повторяет первый.

КЛЮЧЕВОЙ НЮАНС

Различие даже в микрокоде и версии QEMU между хостами ломает миграцию. Мы держим пару хостов на одинаковом релизе LTS и обновляем их последовательно, а не одновременно.



Живая миграция и обслуживание без окна простоя

Что настраиваем

Пара KVM-хостов с общим хранилищем NFS/iSCSI, выделенный migration-VLAN на 10G, доменов 25-40

Как мы это делаем

- 1 Готовим доверие: TLS-сертификаты для libvirt и QEMU-native TLS (default_tls_x509_cert_dir, migrate_tls_x509_verify=1); plain-TCP миграцию не открываем
- 2 В qemu.conf задаём migration_host/migration_address на адрес migration-VLAN, чтобы трафик памяти не шёл через сервисный интерфейс
- 3 CPU-модель: host-model с общим базовым набором флагов пары; совместимость сверяем через virsh cpu-compare с capabilities-XML соседнего хоста
- 4 Миграция: virsh migrate --live --persistent --undefinesource --verbose --tls; на тяжёлых VM добавляем --auto-converge, при большой dirty-rate — post-copy
- 5 Регламент обслуживания: эвакуация всех доменов с хоста, обновление ядра и прошивок, обратный переезд, контроль virsh domstats и steal time сутки после

РЕЗУЛЬТАТ

Плановые работы по железу и обновления ядра перестают быть ночными. VM с 32 ГБ памяти переезжает за десятки секунд, сервисы держат TCP-сессии. Отказ от post-copy на критичных базах сознательный: при обрыве канала он ведёт к потере VM, там работаем pre-copy с auto-converge.

КЛЮЧЕВОЙ НЮАНС

Живая миграция — это не фича гипервизора, а свойство всей связки: одинаковые QEMU, общий storage, отдельный канал и совместимый CPU. Ломается любое из четырёх — ломается всё.



Бэкап и восстановление VM без остановки

Что настраиваем

Хост с 20-40 VM, целевое хранилище — отдельный NAS/репозиторий, RPO сутки для офисных VM и час для баз

Как мы это делаем

- 1 Переводим критичные диски в qcow2 с поддержкой persistent bitmaps — без этого инкрементальный режим недоступен
- 2 Полный слепок — через backup API (virDomainBackupBegin, virsh backup-begin) с созданием чекпойнта; инкременты — от предыдущего чекпойнта по dirty bitmap
- 3 Согласованность гостя: перед снятием дергаем virsh domfsfreeze и domfsthaw через qemu-guest-agent, для СУБД добавляем свой pre/post-скрипт
- 4 Проверка восстановления: раз в месяц поднимаем VM из архива в изолированной сети (libvirt-сеть без элемента <forward>), проверяем загрузку и целостность данных
- 5 Мониторинг: node_exporter + экспортер libvirt, алерты на рост длительности бэкапа, на steal >3% и на iowait гостей >10%

РЕЗУЛЬТАТ

Резервная копия перестаёт быть снимком LVM, снятым вслепую. Инкременты по dirty bitmap занимают минуты вместо часов, VM при этом работает. Тестовое восстановление в изолированной сети выявляет проблемы с загрузчиком и UEFI-переменными до аварии.

КЛЮЧЕВОЙ НЮАНС

Резервная копия, из которой ни разу не поднимали VM, копией не является. Мы закладываем регулярный тест восстановления в регламент обслуживания, а не в благие намерения.



Подводные камни

✗ Оставили NAT-сеть default

ВМ за NAT недоступны из офисной сети, а сама сеть тянет за собой правила фаервола хоста. Мы её удаляем и работаем через bridge.

✗ Балансировка памяти balloon на СУБД

Активный balloon у базы даёт непредсказуемые провалы производительности. Для таких ВМ currentMemory равен memory, баллон не используем.

✗ `cpu mode='host-passthrough'` в паре хостов

Полный проброс модели прибавляет ВМ к конкретному процессору и убивает миграцию на неидентичное железо. По умолчанию берём host-model.

✗ `cache='writeback'` ради скорости

На отказе питания получаем битые файлы гостя. В production ставим `cache='none'` с `io='native'` и полагаемся на BBU контроллера.

✗ Миграция через сервисный интерфейс

Копирование памяти забивает канал управления и мониторинга. Задаём `migration_host/migration_address` на отдельный VLAN.

✗ Открытый 16509 наружу

Незашифрованный libvirtd в периметре недопустим. Работаем только через TLS 16514 или `qemu+ssh`, диапазон миграции держим внутри.

✗ Снапшот вместо бэкапа

Внутренние снимки qcow2 растут, тормозят гостя и живут на том же диске. Это точка отката на час, а не резервная копия.

✗ Нет гостевого агента

Без `qemu-guest-agent` нет корректного выключения, заморозки ФС и адресов гостя. Ставим его в шаблон образа сразу.

Как правильно

МИНИМУМ

- ECC-память, VT-d/AMD-Vi включены, RAID-10 с BBU или mdadm/ZFS
- Ubuntu Server LTS, bridge вместо NAT, autostart для всех доменов
- qemu-guest-agent в каждом госте, virtio-диски и virtio-сеть
- Регулярный бэкап VM на отдельное хранилище и журнал проверок

НОРМАЛЬНО

- Два хоста с общим storage (NFS/iSCSI), рабочая live migration
- LVM thin pool под нагрузку, отдельный dir-пул qcow2 под остальное
- TLS для libvirt и миграции, выделенный migration-VLAN
- Мониторинг: node_exporter + экспортер libvirt, алерты по steal/iowait

ХОРОШО

- Инкрементальный бэкап через backup API и чекпойнты, ежемесячный тест restore
- Hugepages 1 GiB, NUMA-пиннинг и multiqueue virtio для 1C и СУБД
- Вся конфигурация хостов в Ansible, домены в git как XML
- AppArmor/sVirt включён, bond LACP 10G, firewall_backend=nftables (libvirt 10.4+)



Чек-лист самопроверки

- | | |
|---|---|
| ☐ <code>grep -Ес '(vmx svm)' /proc/cpuinfo</code> и <code>kvm-ok</code> дают положительный результат, IOMMU виден в <code>dmesg</code> | ☐ <code>qemu.conf</code> : <code>migration_host</code> на <code>migration-VLAN</code> , диапазон 49152-49215 открыт только между хостами |
| ☐ Сеть <code>default</code> удалена, <code>br0</code> поднят через <code>netplan</code> , <code>libvirt</code> -сеть с <code>forward mode='bridge'</code> в автостарте | ☐ <code>libvirtd</code> и миграция работают только по TLS 16514; <code>plain</code> -порт 16509 закрыт, снаружи — SSH |
| ☐ Пулы хранения определены и в <code>autostart</code> : LVM-пул на NVMe, <code>dir</code> под <code>qcow2</code> , отдельный <code>iso</code> -пул | ☐ <code>qemu-guest-agent</code> установлен, <code>virsh domfsfreeze/domfsthaw</code> отрабатывают на всех ВМ с данными |
| ☐ Диски ВМ: <code>virtio</code> , <code>cache='none'</code> , <code>io='native'</code> , <code>discard='unmap'</code> ; UEFI-ВМ используют OVMF | ☐ Инкрементальные бэкапы идут по чекпойнтам, лог проверяется, копия лежит вне хоста |
| ☐ <code>cpu mode='host-model'</code> , <code>virsh cpu-compare</code> с <code>capabilities</code> соседнего хоста проходит без ошибок | ☐ Раз в месяц восстановление ВМ в изолированной сети с проверкой загрузки и данных |

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.



Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Подбираем и проверяем железо, настраиваем BIOS, RAID и сеть под KVM-хост
- Разворачиваем пару хостов из Ansible-роли: сеть, пулы, TLS, живая миграция
- Мигрируем физические серверы и VM с чужих гипервизоров с планом отката
- Настраиваем инкрементальные бэкапы, мониторинг и регламент тестового restore
- Берём хосты на обслуживание: обновления в рабочее время, эвакуация VM без простоя

15+

лет в ИТ-поддержке

50

рабочих мест — наш профиль

МТС

дата-центр, Москва



КОНТАКТЫ

Обсудить вашу задачу

Сайт **itfresh.ru**

Телефон **+7 903 729-62-41**

Telegram **@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.



itfresh.ru



Техническая база

- 01** libvirt: Domain XML format (formatdomain), диски и CPU (libvirt.org — 2026)
- 02** libvirt: QEMU/KVM hypervisor driver и qemu.conf (libvirt.org — v10.x)
- 03** libvirt kbase: incremental backup, virsh backup-begin (libvirt.org — v7.6+)
- 04** libvirt: Remote support — порты 16509/16514, TLS (libvirt.org — 2026)
- 05** libvirt: Libvirt Daemons — модульные virtqemud/virtnetworkd (libvirt.org — 2026)
- 06** libvirt: network.conf, firewall_backend=nftables (libvirt.org — v10.4+)
- 07** QEMU System Emulation User's Guide, block devices (qemu.org — v8.2)
- 08** Ubuntu Server: пакеты libvirt 10.0.0 и QEMU 8.2.2 (ubuntu.com — 24.04 LTS)
- 09** Внутренний шаблон ITfresh: Ansible-поль kvm-host (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

