Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Ящик Linux-инженера ITfresh 2026: стандарт из 12 утилит

Единый набор утилит на серверах клиентов: ansible-роль, раскладки и работа на инцидентах

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

На серверах клиентов у каждого инженера был «свой» набор: mc у одного, ranger у другого, на третьем сервере вообще пусто. Ночью на инциденте дежурный тратит 15–20 минут на apt install и привыкание вместо диагностики, а на root-разделе копится мусор из чужих утилит. Мы зафиксировали стандарт: 12 утилит, одинаковые алиасы и раскладки, раскатка одной ansible-ролью за 90 секунд.

Почему это важно бизнесу

- Каждая минута простоя 1С/файлового сервера — это остановленная бухгалтерия, склад или клиника; экономим их на старте диагностики
- Предсказуемый инструментарий = любой дежурный работает на любом сервере без «а тут что стоит?»
- Пиннинг версий и контрольные суммы: на проде нет случайных бинарников с чужих репозиториев
- Диагностические инструменты ставятся на время инцидента и снимаются — меньше поверхность атаки
- Runbook в justfile: новый инженер разворачивает и бэкапит проект теми же командами, что и старший

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

12

утилит стандарта: far2l/mc, btop, fzf, ripgrep, fd, eza, bat, just, nushell, zellij, restic, d…

наш стандарт

90 с

раскатка роли itfresh.toolbox на чистый Ubuntu 24.04: apt + pinned-бинарники + bashrc.j2

ansible-core 2.21

<150 МБ

суммарный след всех бинарников на диске — меньше «мусора» от разрозненных установок

наш стандарт

30/12/6

retention restic: --keep-daily 30 --keep-weekly 12 --keep-monthly 6 (+ --keep-hourly 24)

по докам restic 0.19

10%

restic check --read-data-subset=10% еженедельно: проверяем читаемость pack-файлов, не только и…

по докам restic 0.19

2000 мс

update_ms в btop.conf для серверов (дефолт): не грузим CPU слабых VM, при разборе снижаем до 5…

по докам btop 1.4

РЕАЛИЗАЦИЯ 1 ИЗ 3

Раскатка стандарта: ansible-роль itfresh.toolbox

Что настраиваем

Каждый Linux-сервер клиента (Proxmox VM, физика, VPS): Ubuntu 24.04 LTS / Debian 12, учётка администратора не root

Как мы это делаем

1. apt: btop 1.3.0, ripgrep 14.1.0, fzf 0.44.1, bat, fd-find, mc, jq из universe (noble); симлинки batcat→bat и fdfind→fd в /usr/local/bin, чтобы алиасы совпадали с up…
2. GitHub-бинарники pinned в defaults/main.yml: eza 0.23.5, just 1.58.0, nu 0.115.1, zellij 0.45.1, restic 0.19.1, dust 1.2.5, micro 2.0.15; get_url с checksum: sha256…
3. far2l git-снапшот из ppa:far2l-team/ppa (upstream 2.9.0 beta), NetRocks: SFTP/SMB/NFS/WebDAV; на Debian 12 — пакет bookworm; никогда под root
4. template bashrc.j2: алиасы l='eza -la --git', cat→bat, EDITOR=micro, source /usr/share/doc/fzf/examples/key-bindings.bash; update-alternatives --install/--set edito…
5. ansible-lint + molecule на образах noble/bookworm перед мержем; ежеквартальный «технический день» — playbook с тегом toolbox_update

РЕЗУЛЬТАТ

Новый сервер получает полный набор за одну задачу onboarding. Версии одинаковы на всём парке, поэтому justfile и zellij-раскладки переносятся между клиентами без правок, а diff между серверами показывает только данные, не инструменты.

КЛЮЧЕВОЙ НЮАНС

Смешанная стратегия: то, что в apt отстаёт на 1–2 минорных версии, берём из репозитория (безопасность через unattended-upgrades), а быстро развивающиеся утилиты — pinned-бинарником с контрольной суммой.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Рабочее место дежурного: zellij-раскладка server-check

Что настраиваем

Инцидент на сервере 1С/PostgreSQL/MariaDB у клиента: подключение по SSH под admin-учёткой, разбор без установки чего-либо

Как мы это делаем

1. zellij -s itfresh-debug -l /etc/itfresh/zellij/server-check.kdl: панели btop, journalctl -fu <служба>, watch ss -tlnp; повторный вход — zellij attach itfresh-debug
2. rg -z 'OOM|ERROR' /var/log/syslog* -C 5: ключ -z читает ротированные .gz, -t log ограничивает поиск типом *.log, -uu снимает ignore-правила и скрытые файлы
3. dust -d 3 /var и eza -la --sort=size --total-size /var/log: за секунды видно, кто съел диск; fd '\.log$' /var/log -S +500m
4. nu для табличного разбора: ps | where mem > 500MB | sort-by cpu -r | first 10; df -i | detect columns | where {|r| ($r."IUse%" | str trim -c '%' | into int) > 80}
5. session_serialization true в config.kdl (по умолчанию включено): при обрыве SSH раскладка и панели возвращаются командой zellij attach

РЕЗУЛЬТАТ

Инженер за первые 3–5 минут получает картину CPU/RAM/диск/сеть, последние ошибки и открытые порты в одном экране. Ничего не доустанавливается на проде, все команды одинаковы на любом сервере парка.

КЛЮЧЕВОЙ НЮАНС

Zellij перехватывает Ctrl+p/t/n/o/s/h и Alt-комбинации, нужные far2l/mc (Ctrl+O — панели, Alt+F1/F2 — диск); для менеджера переключаемся в locked (Ctrl+g) и держим keybinds в config.kdl.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Бэкап-контур restic + runbook в justfile

Что настраиваем

1С-каталоги, файловые шары и конфиги /etc на серверах клиента; репозиторий на нашем SFTP за пределами площадки клиента

Как мы это делаем

1. RESTIC_REPOSITORY=sftp:itfreshftp:/backup/<client>, RESTIC_PASSWORD_FILE=/etc/restic/password (0600, пароль — в Bitwarden), restic init
2. systemd-timer каждые 6 ч: restic backup <пути> --tag 1c --tag $(hostname) --exclude-file=/etc/restic/excludes --exclude-caches --compression auto
3. Ежедневно restic forget --keep-hourly 24 --keep-daily 30 --keep-weekly 12 --keep-monthly 6 --prune; еженедельно restic check --read-data-subset=10%
4. justfile проекта: backup, restore-test (restic restore latest --target /tmp/restore), clean-logs, deploy version='latest'; set shell := ["bash", "-euo", "pipefail",...
5. Опасные рецепты помечаем [confirm]; set default-list := true — just без аргументов показывает список рецептов, это и есть runbook для новичка

РЕЗУЛЬТАТ

Дедупликация и шифрование на стороне клиента, история за полгода, восстановление на тестовый стенд одной командой. Проверка читаемости данных плановая, а не «когда понадобилось».

КЛЮЧЕВОЙ НЮАНС

restic 0.19 добавил режимы zstd fastest/better (--compression) и ускорил загрузку индекса больших репозиториев; при обрыве бэкапа сначала restic unlock, иначе следующий таймер молча упадёт на stale lock.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **batcat и fdfind вместо bat и fd**
  В Ubuntu/Debian бинарники переименованы из-за конфликта имён; роль ставит симлинки в /usr/local/bin, иначе алиасы и justfile ломаются.

- ✕ **Иконки eza без Nerd Font**
  На терминале клиента без патченного шрифта --icons=always даёт мусор; в алиасах держим --icons=auto и отдельный l без иконок для root.

- ✕ **GitHub из РФ недоступен или медленный**
  get_url падает по таймауту в самый неудобный момент; держим зеркало артефактов и checksum sha256 на каждый файл в defaults роли.

- ✕ **nushell как login shell**
  Ansible, cron и чужие скрипты ждут POSIX-shell; login shell остаётся bash, nu запускаем интерактивно алиасом — только для разбора данных.

- ✕ **apt отстаёт от upstream**
  btop 1.3.0 в noble против 1.4.7 upstream, fzf 0.44 против 0.74; критичные для работы утилиты берём pinned-бинарником, остальное — из apt.

- ✕ **Забытый lock в restic**
  Обрыв SSH во время prune оставляет lock, следующие backup падают; таймер-обёртка делает restic unlock при ошибке 'repository is already locked'.

- ✕ **Диагностика, оставленная на проде**
  strace, tcpdump, bpftrace и pip-инструменты (py-spy в venv) живут в отдельной роли itfresh.toolbox_diag; после инцидента прогоняем её со state: absen…

- ✕ **Файловый менеджер под root**
  F8 в far2l/mc под root на /var/lib — потерянные данные; менеджер только под admin-учёткой, привилегии точечно через sudo.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Единый список утилит и алиасов в bashrc.j2, ставится одной ролью на каждый сервер
- btop, ripgrep, fzf, bat, fd, mc из apt; micro как EDITOR через update-alternatives -…
- restic с retention 30/12/6 и таймером каждые 6 ч, пароль репозитория в сейфе
- Файловые менеджеры и nushell — только под admin-учёткой, не под root

НОРМАЛЬНО

- Pinned-версии GitHub-бинарников с sha256 в defaults роли, зеркало артефактов
- zellij-раскладки server-check.kdl и session_serialization на каждом сервере
- justfile с runbook проекта: backup, restore-test, deploy, clean-logs, [confirm], def…
- restic check --read-data-subset=10% еженедельно, unlock-обёртка в таймере

ХОРОШО

- molecule-тесты роли на noble/bookworm, ansible-lint в CI перед мержем
- Ежеквартальный playbook toolbox_update по всему парку в «технический день»
- Отдельная роль диагностики (strace, tcpdump, bpftrace) с гарантированным удалением
- Тестовое восстановление restic на стенд раз в месяц с фиксацией времени RTO

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] На каждом Linux-сервере компании стоит один и тот же набор утилит, зафиксированный в ansible-роли?
- [ ] Версии бинарников из GitHub закреплены и проверяются контрольной суммой sha256 при установке?
- [ ] Файловые менеджеры (far2l/mc) запускаются только под учёткой администратора, а не под root?
- [ ] Есть zellij/tmux-раскладка для инцидента, которая поднимается одной командой на любом сервере?
- [ ] restic бэкапит 1С, шары и /etc каждые 6 ч с retention 30/12/6 и пароль репозитория лежит в сейфе?
- [ ] Еженедельно выполняется restic check --read-data-subset и ежемесячно тестовое восстановление?
- [ ] У проектов есть justfile с рецептами backup/restore-test/deploy, понятный новому инженеру?
- [ ] Диагностические инструменты (strace, tcpdump, bpftrace) удаляются с прода после инцидента?
- [ ] Роль обновляется по расписанию (квартал), а не «когда кто-то вспомнил»?

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

ЧТО МЫ ДЕЛАЕМ

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).

- Аудит Linux-серверов: что стоит, кем поставлено, сколько занимает — и приведение к единому стандарту
- Разворачиваем роль itfresh.toolbox на ваш парк: apt + pinned-бинарники, алиасы, zellij-раскладки, EDITOR
- Настраиваем restic-контур: SFTP-репозиторий вне площадки, таймеры, retention, check и тестовое восстановление
- Пишем justfile-runbook под ваши сервисы (1С, PostgreSQL, nginx) и обучаем инженеров работе с набором
- Берём серверы на сопровождение: дежурство 24/7 с единым инструментарием и ежеквартальным обновлением

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

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 restic: Removing backup snapshots / Checking integrity / Tuning (restic.readthedocs.io — 0.19.1)

02 Zellij: Layouts (KDL), CLI (--session/--layout) и session_serialization (zellij.dev — 0.45)

03 Nushell Book / Language Reference: filesize, ps, sort-by, detect columns (nushell.sh — 0.115)

04 ripgrep GUIDE: типы файлов (-t), -z, ignore-правила (-u) (github.com/BurntSushi — 15.2.0)

05 Ansible builtin: apt, get_url (checksum), unarchive, template, alternatives (docs.ansible.com — 2.21)

06 far2l README и PPA far2l-team: сборка, NetRocks, SSH (github.com/elfmz — 2.9.0)

07 just Manual: settings (shell, default-list), [confirm], параметры (just.systems — 1.58)

08 Шаблон роли itfresh.toolbox (defaults, bashrc.j2, server-check.kdl) (наш шаблон — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.