Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Kubernetes Deckhouse CSE: кластер с сертификатом ФСТЭК

Как мы разворачиваем DKP Certified Security Edition на bare metal и готовим его к аттестации

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Компания обязана держать ПДн и данные ГИС в аттестованном контуре, а разработка уже живёт в контейнерах. Ванильный Kubernetes аттестовать нечем: сертификата средства контейнеризации по приказу ФСТЭК № 118 у него нет. Мы закрываем это переносом нагрузки на DKP Certified Security Edition — сертифицированную сборку с фиксированным составом.

Почему это важно бизнесу

- Без сертифицированного средства контейнеризации аттестат на ГИС или ИСПДн не выдадут — проект встаёт до замены платформы
- Пересборка кластера «на живую» после замечаний проверки — это простой прод-сервисов и оплата тех же работ второй раз
- Ванильный K8s тянет образы из внешних реестров: в закрытом контуре это вставшие обновления и непропатченные уязвимости
- Своя сборка K8s требует двух инженеров на поддержку; сертифицированная платформа снимает эту постоянную статью затрат
- Без аудита Kubernetes API нечем показать, кто менял секреты и роли: доказательной базы при разборе инцидента нет

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

№ 4860

Сертификат ФСТЭК России от 04.10.2024 на DKP CSE — проверяем срок действия перед стартом

страница DKP CSE, deckhouse.ru

4 класс

4 класс средств контейнеризации и 4 уровень доверия: ГИС и ИСПДн до 1 класса включительно

приказы ФСТЭК № 118 (2022) и № 76 (2020)

1.73.4

Актуальная сертифицированная версия DKP CSE; ставим только сборки из перечня обновлений CSE

перечень обновлений DKP CSE

1.29/1.31

Версии Kubernetes в DKP CSE 1.73; в 1.67 это были 1.27 и 1.29 — апгрейд планируем заранее

перечень обновлений DKP CSE

3 мастера

Кворум etcd: три control-plane узла, иначе перезагрузка одного узла кладёт API кластера

наш стандарт

400 IOPS

Master-узел: 4 CPU, 8 ГБ RAM, 60 ГБ на диске от 400 IOPS: на медленном диске etcd теряет лидера

докам DKP, Going to Production

РЕАЛИЗАЦИЯ 1 ИЗ 3

Базовая установка: bare metal, закрытый контур, dhctl

Что настраиваем

3 master + 2 worker на Astra Linux или РЕД ОС, clusterType: Static, без выхода в интернет

Как мы это делаем

1. Готовим config.yml: ClusterConfiguration (deckhouse.io/v1), clusterType: Static, podSubnetCIDR, serviceSubnetCIDR, podSubnetNodeCIDRPrefix «24» — 254 пода на узел
2. Фиксируем kubernetesVersion явным значением из перечня CSE вместо Automatic: сертифицирована конкретная пара версий платформы и Kubernetes
3. Зеркалируем образы из registry-cse.deckhouse.ru во внутренний реестр: логин license-token, лицензионный ключ из личного кабинета
4. Ставим с отдельной машины контейнером-установщиком: dhctl bootstrap --config, --ssh-user, --ssh-host; на будущем мастере не должно быть docker и containerd
5. Добавляем узлы через NodeGroup, роли frontend и system выносим отдельными группами, чтобы Ingress и мониторинг не жили на мастерах

РЕЗУЛЬТАТ

Кластер поднимается из декларативного конфига за один проход и полностью воспроизводим: тот же config.yml разворачивает идентичный стенд для приёмки. Все образы идут из внутреннего реестра, поэтому обновление не зависит от доступа в интернет.

КЛЮЧЕВОЙ НЮАНС

podSubnetCIDR, serviceSubnetCIDR и podSubnetNodeCIDRPrefix в работающем кластере менять нельзя — только пересоздание. Считаем адресацию до установки и сверяем с сетями офиса.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Контроль нагрузки: политики допуска и сканирование образов

Что настраиваем

Модули admission-policy-engine и operator-trivy на всех прикладных namespace

Как мы это делаем

1. Включаем admission-policy-engine через ModuleConfig и стартуем с podSecurityStandards.enforcementAction: Dryrun, чтобы собрать реальные нарушения
2. Разбираем события, чиним манифесты и только затем переводим enforcementAction в Deny — это значение по умолчанию, но включать его вслепую нельзя
3. Описываем SecurityPolicy на namespace: allowPrivileged: false, allowHostNetwork: false, requiredDropCapabilities, allowedVolumes и seccompProfiles
4. Ставим operator-trivy и в denyVulnerableImages.allowedSeverityLevels оставляем UNKNOWN, LOW, MEDIUM — образы с HIGH и CRITICAL в прод не проходят
5. hostPorts разрешаем точечно: allowedHostPorts — массив диапазонов с min и max под конкретный сервис, а не один диапазон на весь кластер

РЕЗУЛЬТАТ

Под с лишними правами не стартует, а не «стартует и потом обнаруживается на аудите». Отчёты сканера образов и отчёт соответствия FSTEC-21 из operator-trivy ложатся в пакет документов для проверяющего без ручной подготовки.

КЛЮЧЕВОЙ НЮАНС

Deny сразу после установки останавливает половину legacy-приложений. Проходим цикл Dryrun → Warn → Deny и держим согласованный список исключений по namespace.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Наблюдаемость и доказательная база: аудит, логи, алерты

Что настраиваем

Модули runtime-audit-engine и log-shipper плюс встроенный стек мониторинга на system-узлах

Как мы это делаем

1. Включаем runtime-audit-engine: в поде falco, rules-loader, falcosidekick и kube-rbac-proxy; правила описываем ресурсом FalcoAuditRules и держим в git
2. Настраиваем log-shipper: ClusterLoggingConfig на источники и ClusterLogDestination на внешний коллектор — журналы уходят за пределы кластера
3. Включаем аудит Kubernetes API: auditPolicyEnabled в control-plane-manager плюс секрет kube-system/audit-policy, лог в /var/log/kube-audit/audit.log
4. Оповещения заводим ресурсом CustomAlertmanager: отдельные маршруты на дежурного и на владельца сервиса, без единой свалки алертов
5. Ставим резервное копирование etcd по расписанию и раз в квартал проверяем восстановление на отдельном стенде

РЕЗУЛЬТАТ

Есть внешняя копия журналов, независимая от кластера, и проверенный сценарий восстановления control-plane. В аудит-логе видно, кто и когда создавал или менял секрет и роль.

КЛЮЧЕВОЙ НЮАНС

Журналы, лежащие только внутри кластера, теряются вместе с кластером. Приёмник логов ставим вне контура K8s и проверяем доставку тестовым событием, а не по статусу пода.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Ставят EE и надеются переключить потом**
  В CSE нет облачных кластеров и части модулей EE, а переход идёт только между одинаковыми минорными версиями. Сверяем таблицу редакций заранее.
- ✕ **kubernetesVersion: Automatic в контуре**
  Автообновление минорной версии K8s выводит связку за рамки сертифицированной сборки. Пишем версию явно и обновляем осознанно.
- ✕ **Один master «пока для теста»**
  Тестовый кластер незаметно становится продом, а кворума etcd нет. Закладываем три control-plane узла сразу — дорашивать сложнее.
- ✕ **etcd на медленном или общем диске**
  Ниже 400 IOPS etcd начинает терять лидера, API отвечает с задержками. Выделяем мастерам отдельный быстрый диск от 60 ГБ.
- ✕ **enforcementAction: Deny с первого дня**
  Политики валят legacy-нагрузку, и команда отключает модуль целиком. Проходим Dryrun и Warn, копим исключения, потом включаем Deny.
- ✕ **Образы тянутся из внешних реестров**
  В закрытом контуре обновления просто перестают приезжать. Зеркалируем registry-cse.deckhouse.ru во внутренний реестр и следим за наполнением.
- ✕ **Пересчёт подсетей после запуска**
  podSubnetCIDR и serviceSubnetCIDR в работающем кластере не меняются. Пересечение с офисной сетью лечится только пересозданием кластера.
- ✕ **Нет резервной копии etcd**
  Потеря мастеров означает потерю всего состояния кластера. Бэкап etcd по расписанию и проверка восстановления на стенде обязательны.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Сборка DKP CSE только из перечня обновлений, версия K8s задана явно
- Три master-узла, диск под etcd от 400 IOPS, бэкап etcd по расписанию
- Внутреннее зеркало реестра образов, установка без выхода в интернет
- admission-policy-engine включён, SecurityPolicy на прикладных namespace

НОРМАЛЬНО

- enforcementAction: Deny после цикла Dryrun и Warn, исключения согласованы
- operator-trivy на все образы, HIGH и CRITICAL в прод не допускаются
- log-shipper выгружает журналы на внешний коллектор за пределами кластера
- Выделенные группы узлов frontend и system, мониторинг не живёт на мастерах

ХОРОШО

- runtime-audit-engine с правилами FalcoAuditRules в git и маршрутами в SIEM
- Секреты во внешнем хранилище, mTLS для внутреннего трафика платформы
- Ежеквартальные учения по восстановлению control-plane из копии etcd
- Отдельный стенд из того же config.yml под обкатку обновлений платформы

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Действует ли сертификат ФСТЭК на ту версию DKP CSE, которая реально стоит в кластере?
- [ ] Задана ли kubernetesVersion явно или кластер обновляет минорную версию сам?
- [ ] Три ли у нас control-plane узла и держится ли кворум etcd при перезагрузке одного?
- [ ] Проверяли ли мы диск под etcd на 400+ IOPS, а не только на объём?
- [ ] Все ли образы приходят из внутреннего зеркала, а не из внешнего реестра?
- [ ] В каком режиме работает admission-policy-engine — Dryrun, Warn или всё-таки Deny?
- [ ] Сканируются ли образы прода и запрещена ли выкатка с уязвимостями HIGH и CRITICAL?
- [ ] Уходят ли журналы кластера на внешний приёмник, независимый от самого кластера?
- [ ] Восстанавливали ли мы кластер из копии etcd хотя бы раз на отдельном стенде?
- [ ] Есть ли схема адресации подов и сервисов, сверенная с сетями офиса и филиалов?

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).

- Проектируем и разворачиваем DKP CSE на вашем железе: адресация, узлы, реестр, dhctl
- Переносим нагрузку из ванильного Kubernetes или Docker Compose по namespace
- Настраиваем политики допуска, сканирование образов и аудит под требования проверяющего
- Ставим мониторинг, сбор журналов во внешний коллектор и резервное копирование etcd
- Берём кластер на сопровождение: обновления сертифицированных версий и дежурство по алертам

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 Deckhouse Kubernetes Platform Certified Security Edition (deckhouse.ru — 2026)

02 Обновления DKP CSE: перечень сертифицированных версий (deckhouse.ru — 1.73.4)

03 Installing: ClusterConfiguration и параметры кластера (deckhouse.io — v1)

04 Guides: Going to Production, требования к узлам (deckhouse.io — v1)

05 Модуль admission-policy-engine: configuration и CR (deckhouse.io — v1)

06 Модули operator-trivy, runtime-audit-engine, log-shipper (deckhouse.io — v1)

07 Приказы ФСТЭК России № 118 (2022) и № 76 (2020) (fstec.ru — 2022)

08 Наш чек-лист сдачи K8s-кластера в эксплуатацию (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.