

Казбекова Г.Н.¹, Айтбаев Г.Б.²

¹кандидат технических наук, ассоциированный профессор, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави (Казахстан, г.Туркестан.), e-mail: gulnur.kazbekova@ayu.edu.kz

²магистрант, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави (Казахстан, г.Туркестан.), e-mail: gabit.aitbayev@ayu.edu.kz

АРХИТЕКТУРНЫЕ ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ DEVOPS-ИНФРАСТРУКТУРЫ И ИХ АДАПТАЦИЯ ДЛЯ МАЛЫХ И СРЕДНИХ ПРЕДПРИЯТИЙ КАЗАХСТАНА

DEVOPS ИНФРАКҮРЫЛЫМЫН ЖОБАЛАУДЫҢ АРХИТЕКТУРАЛЫҚ ПРИНЦИПТЕРІ ЖӘНЕ ОЛАРДЫҢ ҚАЗАҚСТАНДАҒЫ ШАҒЫН ЖӘНЕ ОРТА БИЗНЕСКЕ БЕЙІМДЕЛУІ

ARCHITECTURAL PRINCIPLES OF DESIGNING DEVOPS INFRASTRUCTURE AND ITS ADAPTATION FOR SMALL AND MEDIUM-SIZED ENTERPRISES IN KAZAKHSTAN

Аннотация. Эффективное проектирование DevOps-инфраструктуры является критическим условием непрерывной, безопасной и предсказуемой эксплуатации современных информационных систем. Для малых и средних предприятий (МСП) Казахстана внедрение DevOps-практик имеет повышенную актуальность вследствие ограниченности бюджета, кадрового состава и инженерной компетенции эксплуатации. Цель исследования — проанализировать архитектурные принципы проектирования DevOps-инфраструктуры и предложить подходы к их адаптации для МСП Казахстана, включая экономически рациональные сценарии развертывания в локальном контуре организации и на виртуальных выделенных серверах (VPS). Рассматриваются вопросы организационной культуры DevOps, автоматизации ключевого контура поставки изменений, архитектуры конвейера непрерывной интеграции и поставки/развертывания (CI/CD), контейнеризации, наблюдаемости и встроенной безопасности (DevSecOps) в условиях ресурсных ограничений. Отдельное внимание уделено управлению секретами, предотвращению конфигурационного дрейфа, обеспечению восстановимости (резервное копирование и проверка восстановления), а также требованиям к минимизации эксплуатационной сложности. Результаты исследования формируют двухуровневую архитектурную модель DevOps-инфраструктуры (базовый и расширенный уровни), предназначенную для повышения управляемости изменений, ускорения релизного цикла и снижения эксплуатационных рисков без опоры на дорогостоящие корпоративные платформы.

Ключевые слова: DevOps, инфраструктура, архитектура, автоматизация, малые и средние предприятия, Казахстан.

Аңдатпа. DevOps-инфрақұрылымды тиімді жобалау заманауи ақпараттық жүйелердің үздіксіз, қауіпсіз және болжамды жұмысын қамтамасыз етудің негізгі шарты болып табылады. Қазақстандагы шағын және орта кәсіпорындар (ШОК) үшін DevOps-тәжірибелерін енгізу ресурстардың (қаржы, кадр, техникалық құзырем) шектеулілігіне байланысты ерекше өзекті. Зерттеудің мақсаты — DevOps-инфрақұрылымды жобалаудың архитектуралық принциптерін талдан, оларды Қазақстандагы ШОК жағдайына бейімдеу тәсілдерін ұсыну, соның ішінде ұйымның жергілікті контурында және виртуалды бөлінген серверлерде (VPS) үнемді орналастыру сценарийлерін қарастыру. Жұмыста DevOps моделнеті, жеткізу конвейерін автоматтандырудың негізгі контуры, үздіксіз интеграция және жеткізу/енгізу (CI/CD) архитектуrasesы, контейнерлеу, бағыламалылық (observability) және қауіпсіздікті өмірлік цикла кіріктіреу (DevSecOps) аспекттері қарастырылған. Құтия деректерді басқару, конфигурациялық дрейфті болдырмай, қалына келтірілмділік (резервтік көшіру және қалына келтіруді тексеру) және операциялық күрделелілікті азайту талаптары арнайы атап етіледі. Нәтижесінде ШОК қажеттілігіне бейімделген DevOps-инфрақұрылымың еki деңгейлі архитектуралық моделі ұсынылады.

Негізгі сөздер: DevOps, инфрақұрылым, архитектура, автоматтандыру, шағын және орта кәсіпорындар, Қазақстан.

Abstract. Effective DevOps infrastructure design is a critical prerequisite for continuous, secure, and predictable operation of modern information systems. For small and medium-sized enterprises (SMEs) in Kazakhstan, DevOps adoption is especially relevant due to constraints in budget, staffing, and operational engineering expertise. This study aims to analyze architectural principles of DevOps infrastructure design and propose adaptation approaches for Kazakhstani SMEs, including cost-efficient deployments in on-premises environments and on virtual private servers (VPS). The research covers DevOps culture, automation of the critical delivery path, CI/CD pipeline architecture,

containerization, observability, and integrated security practices (DevSecOps) under resource constraints. Particular attention is paid to secrets management, configuration drift prevention, recoverability (backup and restore testing), and minimizing operational complexity. The study proposes a two-level DevOps infrastructure model (baseline and advanced levels) designed to improve change controllability, accelerate release cycles, and reduce operational risks without relying on expensive enterprise platforms.

Keywords: DevOps, infrastructure, architecture, automation, SMEs, Kazakhstan.

Введение

Ускорение цифровой трансформации повышает требования к надёжности, безопасности и скорости изменения информационных систем. В этих условиях DevOps — как совокупность организационных практик, инженерных методов и архитектурных решений, направленных на сокращение цикла поставки изменений при сохранении качества — становится одним из базовых подходов к построению жизненного цикла программных систем. На современном этапе DevOps следует рассматривать не только как культурно-процессную модель взаимодействия разработки и эксплуатации, но и как архитектурную рамку, включающую автоматизацию, конвейер поставки изменений (CI/CD), стандартизацию сред, практики наблюдаемости и интеграцию требований информационной безопасности (DevSecOps) [3, 8, 9].

Для малых и средних предприятий (МСП) Казахстана внедрение DevOps-подхода имеет особую значимость. В сравнении с крупными организациями МСП чаще сталкиваются с ограничением финансовых ресурсов, дефицитом квалифицированных инженерных компетенций эксплуатации, совмещением ролей в ИТ-команде, низким уровнем формализации релизных процедур и преобладанием ручных изменений на продуктивных серверах. Эти факторы снижают предсказуемость обновлений, увеличивают риск отказов и удорожают сопровождение. Одновременно усиление конкуренции и рост ожиданий клиентов по доступности и скорости развития цифровых сервисов требуют от МСП перехода к более управляемой модели изменений [1–3].

Несмотря на широкое распространение DevOps в мировой практике, прикладная адаптация его архитектурных принципов к условиям казахстанских МСП в литературе представлена ограниченно. Значительная часть существующих подходов ориентирована на высокую зрелость процессов и доступ к развитым платформам, тогда как для МСП типичны сценарии эксплуатации на собственных серверах и VPS, необходимость минимизировать эксплуатационную сложность и зависимость от внешних поставщиков, а также ограниченные возможности по внедрению комплексных корпоративных инструментов [3, 7, 12, 14]. Это формирует практическую задачу: выделить инвариантные (независимые от конкретных инструментов) архитектурные принципы DevOps и предложить реализуемую поэтапную модель для МСП Казахстана.

Цель исследования — провести анализ архитектурных принципов DevOps-инфраструктуры и предложить подходы к их адаптации для малых и средних предприятий Казахстана, с фокусом на рациональном базовом контуре и траектории дальнейшего развития.

Методы исследования

Дизайн исследования

Применён комплексный дизайн, включающий:

- систематизированный обзор публикаций и отраслевых материалов;
- сравнительный анализ архитектурных альтернатив по фиксированным критериям;
- архитектурное моделирование (логическая модель и модель развёртывания);
- экспертно-аналитическую оценку ограничений МСП (кадровых, финансовых, эксплуатационных);

- прикладной сценарий развёртывания в среде VPS с использованием измеримых эксплуатационных и процессных показателей.

Протокол обзора литературы

Источники: IEEE Xplore, ACM Digital Library, Scopus, Web of Science, а также отраслевые публикации и локальные материалы для уточнения контекста Казахстана.

Поисковые запросы: «DevOps AND SMEs», «DevSecOps AND CI/CD AND small medium enterprises», «delivery pipeline architecture AND resource constraints», «observability AND incident response AND small teams», «infrastructure as code AND SMEs», а также русскоязычные запросы по автоматизации релизов, DevOps в малом бизнесе и эксплуатации на VPS.

Период: 2020–2025 гг.

Критерии включения: материалы с описанием DevOps/DevSecOps практик и архитектуры CI/CD, инфраструктуры как кода, контейнеризации, наблюдаемости; кейсы и исследования с явными ограничениями малых команд.

Критерии исключения: публикации без контекста внедрения; материалы с выводами, непереносимыми на ограниченные среды; дубли.

По результатам отбора выявлено ограниченное число работ, ориентированных на практическую адаптацию DevOps для МСП: большинство публикаций фокусируются на барьерах и сложностях внедрения, тогда как методически оформленных адаптационных моделей существенно меньше (что подтверждает актуальность темы).

Метод сравнительного анализа архитектур

Сравнение вариантов архитектуры проведено по группе критериев (стоимость внедрения, стоимость владения, воспроизводимость, управляемость отката, безопасность управления секретами, наблюдаемость, требования к компетенциям, масштабируемость, риск ручных изменений). Для представления результата использована шкала 1–5 (1 — наименее предпочтительно/наиболее рискованно; 5 — наиболее предпочтительно/наименее рискованно). Оценки интерпретируются как экспертно-аналитические и предназначены для поддержки выбора «минимально достаточной» архитектуры в условиях МСП.

Анализ и результаты

Архитектурные принципы DevOps-инфраструктуры (инвариантные к инструментам)

1. Управляемость изменений и единый источник истины

Изменения программного кода, конфигураций и инфраструктуры должны быть трассируемыми (кто/что/когда/почему) и воспроизводимыми из версионируемого источника. Цель — устранение неконтролируемых ручных вмешательств в продуктивной среде как системного риска.

2. Воспроизводимость и стандартизация сред

Среды разработки, тестирования и эксплуатации должны быть согласованы по принципам конфигурации и процедурам обновления. В контексте МСП приоритетом является стабильная воспроизводимость сценариев развёртывания и обновления, а не максимальная сложность автоматизации.

3. Автоматизация критического контура поставки изменений

Автоматизация должна охватывать критический путь: сборка → проверка → формирование артефакта → развёртывание → пост-проверка → контролируемый откат. Такой подход позволяет повысить предсказуемость релизов без избыточной автоматизации второстепенных операций.

4. Наблюдаемость и управляемая обратная связь

Наблюдаемость определяется как минимально достаточный набор метрик и журналов событий, обеспечивающий подтверждение корректности релиза, выявление деградаций до

инцидента и сокращение времени диагностики. Для МСП рационален ограниченный набор сигналов, привязанный к эксплуатационным рискам и доступным компетенциям.

5. Встроенная безопасность (DevSecOps)

Практики безопасности должны быть частью архитектуры и конвейера поставки изменений: управление секретами, разграничение доступа по принципу наименьших привилегий, базовые проверки уязвимостей и безопасные процедуры обновления. Это снижает вероятность инцидентов, связанных с человеческим фактором и техническим долгом.

6. Восстановимость и контролируемый откат

Откат прикладных компонентов должен быть штатной процедурой. Данные и файловый контент (состояние системы) требуют отдельного контура защиты: резервное копирование, регламент тестирования восстановления, определение реалистичных целевых показателей восстановления для МСП (например, на уровне регламентируемых сроков восстановления и допустимой потери данных).

Контекст Казахстана: типовые ограничения МСП и архитектурные следствия

Для МСП Казахстана характерны: ограниченный бюджет; совмещение ролей (разработчик одновременно выполняет задачи эксплуатации); дефицит компетенций в области инженерии надёжности; размещение систем на VPS или собственных серверах; высокая доля ручных операций через SSH; отсутствие измеримых показателей доступности и мониторинга.

Архитектурное следствие: приоритетом является не копирование корпоративных (enterprise) практик, а формирование минимально достаточного, устойчивого контура, в котором ключевыми являются управляемость изменений, разделение приложения и состояния системы, наблюдаемость и проверяемая восстановимость.

Таблица 1. Ограничения и архитектурные следствия

Типовое ограничение	Следствие для архитектуры
Малые команды, совмещение ролей	Минимизация эксплуатационной сложности, стандартизованные процедуры
VPS/локальные серверы как базовая среда	Простые схемы развёртывания, минимизация внешних зависимостей
Нестабильность релизных процедур	Формализация регламента релиза и отката, воспроизводимые сценарии
Ручные правки на сервере	Единый источник истины, исключение конфигурационного дрейфа
Ограниченный бюджет на платформенные решения	Опора на самостоятельно администрируемые решения и ПО с открытым исходным кодом
Слабая наблюдаемость	Введение базовых метрик и оповещений (доступность, ошибки, ресурсы)
Рост требований безопасности	Секреты вне репозитория, разграничение доступа, базовые проверки уязвимостей
Отсутствие дисциплины резервного копирования	Регламент, хранение копий вне основного узла, обязательное тестирование восстановления

Сравнительный анализ архитектурных вариантов

Варианты сравнения:

- A1: развёртывание через SSH и скрипты, преимущественно ручные операции, без контейнеризации.
- A2: контейнеризация и развёртывание на VPS с использованием Docker Compose.
- A3: контейнерная оркестрация (кластерный подход) как вариант для роста зрелости.

Таблица 2. Матрица сравнения вариантов (шкала 1–5)

Критерий	A1: ручное развёртывание	A2: Compose на VPS	A3: оркестрация	Комментарий для МСП
Стоимость внедрения	5	4	2	A1 дешевле в начале, но повышает эксплуатационные риски
Стоимость владения	3	4	3	A2 даёт лучший баланс для малых команд
Воспроизводимость	1	4	5	Контейнеризация существенно повышает воспроизводимость
Контролируемый откат	2	4	5	Версионирование образов ускоряет откат
Управление секретами	1	4	5	В A2 реализуем дисциплинированный контур секретов
Наблюдаемость	2	4	5	Рост зрелости требует усиления мониторинга
Требования к компетенциям	4	3	2	Оркестрация часто требует выделенной экспертизы
Масштабируемость	1	3	5	A2 ограниченно масштабируем, но этого часто достаточно для МСП
Риск ручных вмешательств	1	3	4	Снижается при использовании версионирования и стандартизации

Вывод: для большинства МСП Казахстана архитектура A2 (Docker Compose на VPS) является наиболее рациональным базовым вариантом: она повышает управляемость и воспроизводимость при умеренных требованиях к компетенциям и эксплуатационной нагрузке. Архитектура A3 оправдана при росте нагрузки и команды при наличии компетенций и экономической целесообразности.

Двухуровневая адаптивная архитектура DevOps для МСП Казахстана

I) Базовый уровень (минимально достаточный контур для VPS/локальных серверов)

Цель: обеспечить управляемость изменений, воспроизводимость, предсказуемый релиз и откат, минимальную наблюдаемость и безопасную работу с секретами при минимальной эксплуатационной сложности.

Обязательные компоненты базового уровня

1. *Репозиторий версий (Git)* как единый источник истины для кода, конфигурации и описаний развёртывания.
2. *Разделение приложения и состояния системы*: данные СУБД и файловый контент размещаются в томах хранения; вычислительные компоненты — как контейнеры из версионируемых образов.
3. *Сборка артефакта в CI*: формирование контейнерного образа с уникальным тегом версии.
4. *Реестр контейнерных образов*: локальный или внешний (для обеспечивающего отката и повторяемости поставки).
5. *Развёртывание на сервере по неизменяющемуся сценарию*: единая инструкция и единый порядок команд; минимизация ручных вариаций.
6. *Секреты вне репозитория*: переменные окружения/файл конфиденциальных параметров с ограничением прав доступа; запрет хранения секретов в системе контроля версий.
7. *Резервное копирование состояния системы*: дампы СУБД и архивирование файлового контента с хранением копий вне основного узла.
8. *Наблюдаемость базового уровня*: показатели ресурсов узла (CPU/RAM/диск/сеть), доступность сервиса, коды ошибок (в том числе 4xx/5xx), прикладные журналы и журналы СУБД в объёме, достаточном для диагностики.
9. *Откат*: возврат к предыдущему известному рабочему образу; для изменений схемы БД — регламент совместимости и точки восстановления.

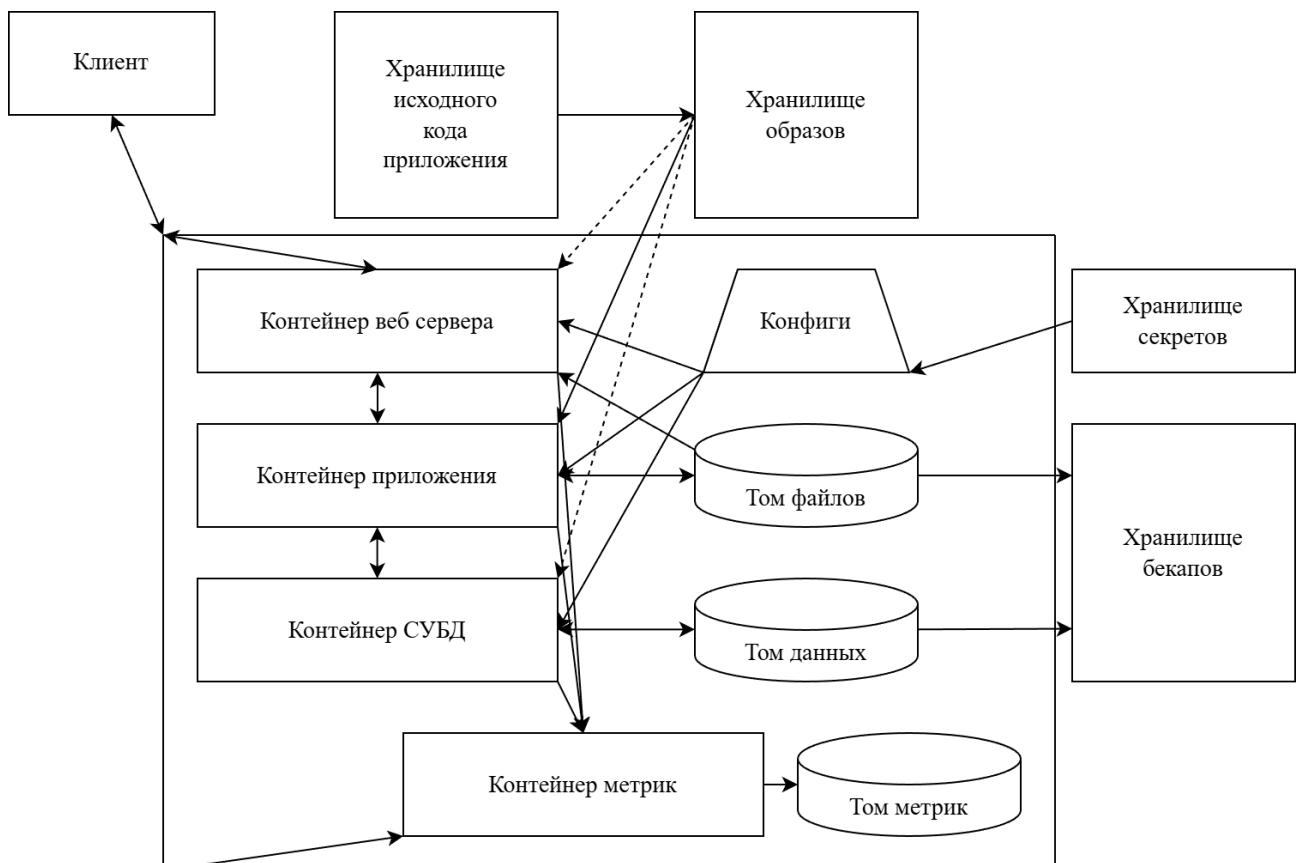


Схема 1. Логика развёртывания базового уровня

2) Расширенный уровень (рост зрелости, усиление контроля и безопасности)

Цель: повысить качество релизов, безопасность и управляемость при увеличении нагрузки, критичности сервиса или численности команды.

Добавляемые практики и элементы

1. *Политики управления изменениями*: защищённые ветви, обязательное рецензирование кода, запрет прямых изменений на продуктивной среде без процедуры изменения.
 2. *Усиление DevSecOps*: статический анализ кода на уязвимости, анализ зависимостей, контроль контейнерных образов, регулярные обновления, минимизация базовых образов.
 3. *Сегментация сети и доступа*: закрытие СУБД от внешнего доступа, принцип наименьших привилегий, усиление аутентификации и контроля административных доступов.
 4. *Наблюдаемость расширенного уровня*: метрики приложения (ошибки, задержки, показатели производительности), централизованные журналы, оповещения на основе целевых показателей уровня сервиса (SLO), при необходимости — трассировка запросов.
 5. *Инфраструктура как код*: формализованное описание инфраструктуры и конфигураций для исключения конфигурационного дрейфа и ускорения миграции/восстановления.
 6. *План обеспечения непрерывности*: формализация целевых параметров восстановления, регулярные тесты восстановления, сценарии отказа провайдера и деградации хранилища.

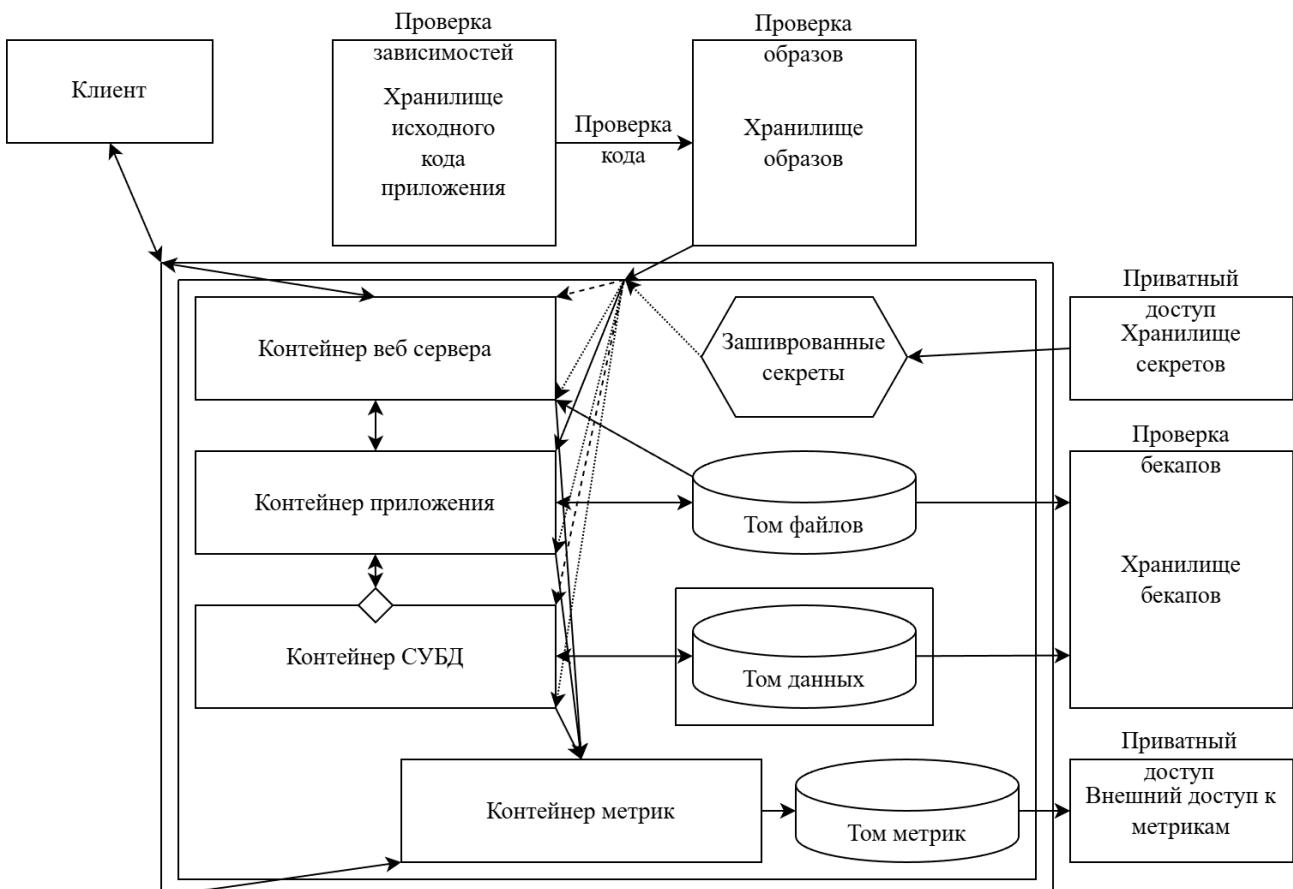


Схема 2. Логика развёртывания расширенного уровня

Риски, ограничения и границы применимости

Таблица 3. Риски и меры контроля

Риск/ограничение	Проявление	Контроль/мера
Недостаток компетенций	Срыв внедрения из-за избыточной сложности	Начинать с базового уровня, унифицировать инструкции, чек-листы, минимальный стек
Попадание секретов в репозиторий	Компрометация доступов	Проверки перед фиксацией/в конвойере, ротация секретов, секреты вне репозитория
Резервные копии без восстановления	Ложное ощущение защищённости	Регламент и обязательные тесты восстановления
Неконсистентная контейнеризация	Рост сложности сопровождения	Стандартизировать Compose-описания, разделить состояние и вычисления, зафиксировать процедуры
Зависимость от одного узла	Единая точка отказа	Внешнее хранение копий, план миграции, сценарии восстановления
Отсутствие измеримых показателей	Невозможность доказать эффект	Введение процессных и эксплуатационных показателей, журналирование изменений и инцидентов

Границы применимости: базовый уровень ориентирован на типовые веб-системы МСП (сайты услуг, электронная коммерция, внутренние порталы), где допустима эксплуатация на одном сервере при наличии восстановимого контура данных. Для высоконагруженных и критичных к простоям систем требуется расширенный уровень и/или кластерная архитектура.

Прикладной сценарий внедрения базового уровня в среде VPS

Исходные условия

Система: сервисный сайт/интернет-магазин на CMS или аналогичное веб-приложение.

Инфраструктура: один VPS; веб-компонент, прикладной компонент, СУБД, файловый контент (загрузки).

Проблемы: ручные развёртывания по SSH; изменения «на месте»; непредсказуемость обновлений; слабый откат; нерегулярные резервные копии без проверки восстановления; отсутствие метрик доступности.

Реализация базового уровня

- Сервисы описаны как контейнеры (веб-уровень, приложение, СУБД).
- Состояние системы вынесено в тома хранения (данные СУБД и файловый контент).
- Секреты размещены во внешнем контуре конфиденциальных параметров с ограничением доступа.
- Введён минимальный конвойер поставки: сборка образа → размещение в реестре → развёртывание на сервере по стандартизированному сценарию.
- Настроены базовые показатели наблюдаемости: ресурсы узла и ошибки сервиса (включая 5xx).

- Организован контур резервного копирования: дамп СУБД и архив файлового контента с хранением вне основного узла; введён регламент проверки восстановления.
- Определён механизм отката: возврат к предыдущей версии образа; при необходимости — восстановление состояния системы согласно регламенту.

Таблица 4. Пример показателей эффекта внедрения базового уровня

Показатель	До внедрения	После внедрения	Метод измерения
Время цикла изменений	Часы/дни, высокая вариативность	Десятки минут, предсказуемо	Время от фиксации изменений до развёртывания в продуктивной среде
Частота релизов	Низкая из-за высокого риска	Повышается за счёт управляемого отката	Количество релизов за фиксированный период
Доля проблемных релизов	Выше из-за ручных операций	Ниже за счёт воспроизводимости	Доля релизов с инцидентом/откатом
Среднее время восстановления	Часы	Минуты–десятки минут	Журнал инцидентов, время до восстановления сервиса
Объём ручных операций	Высокий	Снижён и регламентирован	Количество ручных шагов по инструкции развёртывания
Конфигурационный дрейф	Высокий	Низкий при соблюдении регламента	Сверка конфигураций, соответствие описанию развёртывания
Наблюдаемость	Практически отсутствует	Введены метрики и оповещения	Доступность, ошибки, ресурсы узла, журналы

Обсуждение

Сопоставление архитектурных вариантов показывает, что для МСП Казахстана определяющим является баланс между управляемостью изменений и эксплуатационной сложностью. DevOps-принципы целесообразно реализовывать не как полный «корпоративный» контур, а как минимально достаточную архитектуру, обеспечивающую снижение рисков при ограниченных ресурсах.

Наиболее устойчивый эффект в условиях МСП обеспечивают:

- устранение ручных изменений в продуктивной среде и фиксация описания развёртывания как версионируемого контура;
- контейнеризация вычислительных компонентов и разделение приложения и состояния системы;
- стандартизованный сценарий релиза с обязательной пост-проверкой;
- штатный механизм отката к предыдущей версии;
- резервное копирование с обязательной проверкой восстановления;

- базовая наблюдаемость, ориентированная на эксплуатационные риски и доступные компетенции.

Поэтапная траектория «базовый уровень → расширенный уровень» снижает вероятность организационного срыва внедрения и позволяет получать измеримые улучшения без опоры на дорогостоящие платформы и без перегрузки команды.

Заключение

В исследовании сформировано целостное представление о возможностях и ограничениях внедрения DevOps-инфраструктуры в малых и средних предприятиях Казахстана. Показано, что ключевые проблемы МСП связаны с ограниченностью ресурсов, низкой формализацией релизных процедур, дефицитом компетенций эксплуатации и преобладанием ручных операций. На этой основе выделены инвариантные архитектурные принципы DevOps-инфраструктуры: управляемость изменений и единый источник истины, воспроизводимость и стандартизация сред, автоматизация критического контура поставки изменений, наблюдаемость и обратная связь, встроенная безопасность (DevSecOps), восстановимость и контролируемый откат [1, 3–5, 7, 12, 14].

Предложена двухуровневая архитектурная модель (базовый и расширенный уровни), ориентированная на практическую реализуемость в типовых для МСП сценариях эксплуатации на собственных серверах и VPS. Модель обеспечивает повышение управляемости изменений, сокращение времени вывода обновлений, снижение эксплуатационных рисков и укрепление безопасности жизненного цикла изменений при сохранении рациональных затрат. Результаты могут быть использованы как методическая основа для проектирования DevOps-инфраструктуры в МСП Казахстана и дальнейшего развития практико-ориентированных рекомендаций по цифровой трансформации бизнеса.

Список использованной литературы

1. Sanjeetha, M. B. F., & et al. (2023). Development of an alignment model for the implementation of DevOps in SMEs: An exploratory study. IEEE Access, 11, 144213–144225. <https://doi.org/10.1109/ACCESS.2023.3344040>
2. Tennakoon, H., & Jayatissa, Y. (2025). Achieving sustainability in Sri Lankan SME IT firms through DevOps. In 2025 International Research Conference on Smart Computing and Systems Engineering (SCSE). IEEE. <https://doi.org/10.1109/SCSE65633.2025.11031052>
3. Krey, M. (2022). DevOps adoption: Challenges & barriers. In Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS) (pp. 7297–7309). <https://doi.org/10.24251/HICSS.2022.877>
4. Cheenepalli, J., Hastings, J. D., Ahmed, K. M., & Fenner, C. (2025). Advancing DevSecOps in SMEs: Challenges and best practices for secure CI/CD pipelines. In 2025 13th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1–6). IEEE. <https://doi.org/10.1109/ISDFS65363.2025.11011960>
5. Singh, S. (2025). Secure software development life cycle: Implementation challenges in small and medium enterprises (SMEs). Authorea (Preprint). <https://doi.org/10.22541/au.174585836.63395541/v1>
6. Trush, A. V. (2023). Автоматизированное управление жизненным циклом программного обеспечения на основе проектов компании «Технопарк-Автоматизация» (Master’s thesis). Уральский федеральный университет имени первого Президента России Б. Н. Ельцина.
7. Малышев, М. О., & Чулюков, Н. В. (2025). Оценка зрелости процессов управления инновационными проектами в малых ИТ-компаниях. International Journal of Humanities and Natural Sciences, (5-2), 358–364. <https://doi.org/10.24412/2500-1000-2025-5-2-358-364>
8. Ермаков, А. С. (2020). Перспективное развитие методологии DevOps. Вестник НГУЭУ, (4), 174–183. <https://doi.org/10.34020/2073-6495-2020-4-174-183>
9. Линская, А. Н. (2022). Характеристики и описание DevOps. StudNet, 5(1), 339–343.
10. Коробка, М. (2025). Почему малому бизнесу важно инвестировать в DevOps? Doers Doings.
11. Коробка, М. (2025). Сравнение подходов DevOps в России и Европе: что работает лучше. TProger.
12. Astana Hub. (2024). Секреты успешного применения DevOps в малом бизнесе. Astana Hub.
13. Radcop Online. (2025). Безопасная разработка ПО в малом и среднем бизнесе (МСБ): практическое руководство DevSecOps. Radcop Online.

14. Бекжанов, Д. (2025). Почему аутсорсинг DevOps услуг — ключ к успеху компаний. Digital Business Kazakhstan.
15. KEENCOMPUTER. (2025). Accelerating SME ecommerce and website growth through launch frameworks, digital transformation, and DevOps (White paper). Keen Computer Solutions.

References

1. Sanjeetha, M. B. F., & et al. (2023). Development of an alignment model for the implementation of DevOps in SMEs: An exploratory study. *IEEE Access*, 11, 144213–144225. <https://doi.org/10.1109/ACCESS.2023.3344040>
2. Tennakoon, H., & Jayatissa, Y. (2025). Achieving sustainability in Sri Lankan SME IT firms through DevOps. In *2025 International Research Conference on Smart Computing and Systems Engineering (SCSE)*. IEEE. <https://doi.org/10.1109/SCSE65633.2025.11031052>
3. Krey, M. (2022). DevOps adoption: Challenges & barriers. In *Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS)* (pp. 7297–7309). <https://doi.org/10.24251/HICSS.2022.877>
4. Cheenepalli, J., Hastings, J. D., Ahmed, K. M., & Fenner, C. (2025). Advancing DevSecOps in SMEs: Challenges and best practices for secure CI/CD pipelines. In *2025 13th International Symposium on Digital Forensics and Security (ISDFS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ISDFS65363.2025.11011960>
5. Singh, S. (2025). Secure software development life cycle: Implementation challenges in small and medium enterprises (SMEs). *Authorea (Preprint)*. <https://doi.org/10.22541/au.174585836.63395541/v1>
6. Trush, A. V. (2023). *Avtomatizirovannoe upravlenie zhiznennym tsiklom programmного обеспеченiya na osnove proektor kompanii "Tekhnopark-Avtomatizatsiya"* (Master's thesis). Ural'skii federal'nyi universitet imeni pervogo Prezidenta Rossii B. N. El'tsina.
7. Malyshev, M. O., & Chulyukov, N. V. (2025). Otsenka zrelosti protsessov upravleniya innovatsionnymi proektami v malykh IT-kompaniyakh. *International Journal of Humanities and Natural Sciences*, (5-2), 358–364. <https://doi.org/10.24412/2500-1000-2025-5-2-358-364>
8. Ermakov, A. S. (2020). Perspektivnoe razvitiye metodologii DevOps. *Vestnik NGUEU*, (4), 174–183. <https://doi.org/10.34020/2073-6495-2020-4-174-183>
9. Linskaya, A. N. (2022). Kharakteristiki i opisanie DevOps. *StudNet*, 5(1), 339–343.
10. Korobka, M. (2025). Pochemu malomu biznesu vazhno investirovat' v DevOps? *Doers Doings*.
11. Korobka, M. (2025). Sravnenie podkhodov DevOps v Rossii i Evrope: chto rabotaet luchshe. *TProger*.
12. Astana Hub. (2024). Sekrety uspeshnogo primeneniya DevOps v malom biznese. *Astana Hub*.
13. Radcop Online. (2025). Bezopasnaya razrabotka PO v malom i sredнем biznese (MSB): prakticheskoe rukovodstvo DevSecOps. *Radcop Online*.
14. Bekzhanov, D. (2025). Pochemu autsorsing DevOps uslug — klyuch k uspekhu kompanii. *Digital Business Kazakhstan*.
15. KEENCOMPUTER. (2025). Accelerating SME ecommerce and website growth through launch frameworks, digital transformation, and DevOps (White paper). Keen Computer Solutions.

Авторлар туралы мәліметтер

№	Аты-жөні, ғылыми дәрежесі, жұмыс немесе оқу орны, қала, мемлекет, автордың e-mail мекенжайы және үялы телефон номірі.
1	Казбекова Г. – техника ғылымдарының кандидаты, қауымдастырылған профессор, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан қ., Қазақстан, e-mail: gulnur.kazbekova@ayu.edu.kz , ORCID: 0000-0002-2756-7926, +77751333354
	Kazbekova G. - Candidate of Technical Sciences, Associate Professor, Ahmet Yassawi University, Turkestan, Kazakhstan, e-mail: gulnur.kazbekova@ayu.edu.kz , ORCID: 0000-0002-2756-7926, +77751333354
	Казбекова Г. – кандидат технических наук, ассоциированный профессор, Международный казахско-турецкий университет им. Ходжи Ахмеда Ясави, г. Туркестан, Казахстан, e-mail: gulnur.kazbekova@ayu.edu.kz , ORCID: 0000-0002-2756-7926, +77751333354
2	Айтбаев Г. – магистрант, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан қ., Қазақстан, e-mail: gabit.aitbayev@ayu.edu.kz , +77074162928
	Aitbayev G. - master's student, Ahmet Yassawi University, Turkestan, Kazakhstan, e-mail: gabit.aitbayev@ayu.edu.kz , +77074162928
	Айтбаев Г. – магистрант, Международный казахско-турецкий университет им. Ходжи Ахмеда Ясави, г. Туркестан, Казахстан, e-mail: gabit.aitbayev@ayu.edu.kz , +77074162928