“O‘ZBEKISTON SANOAT-QURILISH BANKI” AKSIYADORLIK TIJORAT BANKI
АКЦИОНЕРНЫЙ КОММЕРЧЕСКИЙ БАНК “УЗБЕКСКИЙ ПРОМЫШЛЕННО-СТРОИТЕЛЬНЫЙ БАНК

TEXNIK TOPSHIRIQNOMA

№ 1990
2025 yil «19» «iyun»

ТЕХНИЧЕСКОЕ ЗАДАНИЕ
Внедрения оркестратора Kubernetes

Ташкент 2025 г.

1. Общие сведения

1.1 Полное наименование ИС

Полное наименование системы: Информационная система управления контейнерной оркестрацией на базе Kubernetes.

1.2 Наименование организаций заказчика

Заказчик – АКБ «Узпромстройбанк»

Адрес «Заказчика»: Республика Узбекистан, г.Ташкент, 100000, Юнусабадский район, ул. Шахрисабзская, дом №3; Тел.: (998-78) 777 77 55 (7054)

МФО: 00440; ИНН: 200 833 707;

Наименование банка: ОПЕРУ при АКБ «Узпромстройбанк»

Адрес электронной почты: info@sqb.uz;

1.3 Перечень документов, на основании которых создается ИС

Текущие потребности Банка.

1.4 Плановые сроки начала и окончания работ

В рамках реализации данного проекта следует учитывать дорожную карту и реализовывать ее частями. Необходимо внедрить данное решение и процессы к нему в течение 3 месяцев и далее обеспечить техническую поддержку не менее 4 месяцев с функцией передачи знаний внутренней команды Банка;

Сроки реализации проекта – не должны превышать 180 календарных дней.

1.5 Требования к Исполнителю

В штате исполнителя необходимо наличие 5 сертифицированных специалистов, с предоставлением подтверждающих документов (сертификатов CKAD);

Иметь опыт аналогичных, реализованных проектов в Банках Узбекистана.

2. Назначение и цели создания ИС

2.1 Назначение ИС

В рамках ИТ стратегии Банка необходимо подготовить, настроить и запустить весь процесс разработки и доставки приложений во все наши контуры. Это даст возможность контролировать выполнение обновлений и обеспечить максимальный time 2 market наших приложений.

Основные цели внедрения оркестратора Kubernetes:

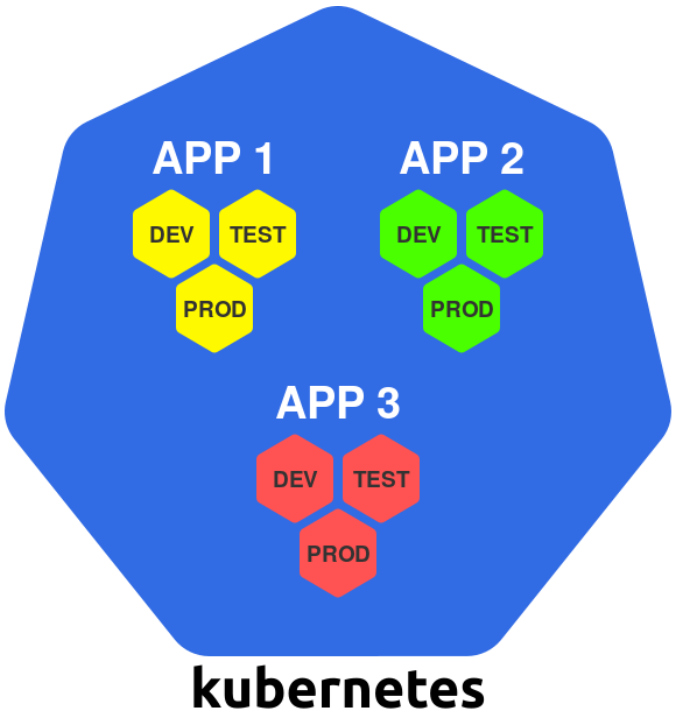
- обеспечить масштабируемость всех приложений;
- повышение отказоустойчивости приложений;
- централизованное хранилище логов приложения и возможность передачи их в SIEM банка (только аудит логов);
- обеспечение безопасности доставки обновления без необходимости доступа в продуктивный сегмент разработчиков и менеджеров;

- повышение observability приложений за счет мониторинга всех процессов и инфраструктуры;

3. Общие требования для окружений

3.1 Концептуальная архитектура кластеров

Необходимо использовать подход “один кластер Kubernetes на каждое окружение”. Данный подход обеспечит для нас изолированность продуктивной среды от всех остальных, дает возможность ограничивать доступ для разных окружений.



3.2. Состав сред и их назначения

В банке присутствует четыре среды:

DEV	Полностью нестабильное окружение для разработчиков
UAT	Тестирование функционала бизнес заказчиком (стабильная версия). В нашей концепции это будет еще и некий PREPROD
TEST	Проведение тестов (не стабильная версия)
PROD	Ограниченный по доступу контур (стабильная версия)

3.3. Функциональные требования к окружениям

- каждое окружение в банке имеет собственный кластер оркестровки на Kubernetes (см. п. 3.1);
- продуктивная среда имеет резервную копию;

- все кластеры для среды DEV, TEST, UAT, PROD должны быть изолированы друг от друга и находиться в различных VLAN;
- все среды развернуты в нашем ЦОД (включая master, compute, NFS и т.д.);
- обязательное использование встроенного vault для хранения секретов;
- использовать только технологию Helm для размещения инкрементов приложений и обновления кластеров;
- в качестве сетевого плагина использовать calico с возможностью установки network policy;

4. Требование к логированию кластеров

В рамках построения корпоративной информационной инфраструктуры и обеспечения должного уровня операционной прозрачности, кластер Kubernetes должен иметь централизованную, масштабируемую и гибко настраиваемую систему логирования. Логирование является критически важным элементом эксплуатации, поддержки, анализа инцидентов и обеспечения информационной безопасности.

4.1. Централизованный сбор логов

Логирование должно быть реализовано в виде централизованной архитектуры с использованием унифицированной системы агрегации логов. В качестве механизма экспорта логов на стороне кластеров должен использоваться компонент Fluentd (или его совместимый аналог), обеспечивающий сбор, фильтрацию и маршрутизацию лог-сообщений с приложений и инфраструктурных компонентов. Сбор осуществляется на уровне родов, нод, ingress-контроллеров и вспомогательных сервисов. В качестве хранилища и интерфейса для анализа логов должна использоваться высокопроизводительная и масштабируемая платформа OpenSearch, обеспечивающая полнотекстовый поиск, агрегации и визуализацию логов.

4.2. Структурированный формат логов

Все логи, поступающие в централизованную систему, должны быть представлены в структурированном формате JSON. Это необходимо для обеспечения машинно читаемости, возможности гибкого поиска, построения дашбордов, триггеров на события, а также последующего анализа с применением инструментов машинного обучения и корреляции событий. Так же следует учитывать, что данные логи будут направлять в SIEM систему банка для дальнейшего анализа (используется IBM QRADAR).

4.3. Временные метки и синхронизация

Каждое лог-сообщение, независимо от источника, должно содержать обязательную временную метку в формате UTC, согласованную с системным временем и синхронизированную по средствам службы времени (например, NTP). Это необходимо для точного упорядочивания событий, особенно в случае инцидентов, затрагивающих несколько компонентов одновременно и разбора форс-мажорных ситуации.

4.4. Поддержка уровней логирования

Все кэшируемые события должны иметь обозначенные уровни логирования, соответствующие промышленным стандартам, таким как **ERROR, WARN, INFO, DEBUG**, TRACE. Необходимо обеспечить возможность настройки детальности логов в зависимости от среды и текущих задач: от сокращённого аудита в PROD до расширенной отладки в DEV/UAT средах.

Уровень	Описание	Пример
ERROR	Регистрация серьезных ошибок, которые приводят к нарушениям SLA. Это могут быть ошибка логики, обработки данных, инфраструктурные проблемы и т.д.	{"level":"ERROR","timestamp":"2025-04-15T10:00:00Z","message":"Failed to process payment","error":"NullPointerException","service":"iABS has no connection"}
WARN	События, которые не прерывают выполнение, но указывают на потенциальную проблему или аномалию. Часто возникают и не всегда приводят к остановки системы	Не найден файл, но при этом процесс работы приложения продолжается
INFO	Фиксируют стандартное, ожидаемое поведение системы - важные этапы бизнес- и системных процессов.	Информирует о тех или иных событиях в системах. INFO могут быть как системными/пользовательскими, так и аудит логами
DEBUG	Используется во время разработки и тестирования	Нужно для проверки функционал.

4.5. Список компонентов, подлежащих обязательному логированию

Базы данных: RDBMS, NoSQL, Time Series DB, Графовые/Векторные базы данных и файловые/NFS-хранилища или Object Storage (Ceph, MinIO, S3).

Системные компоненты кластера: kube-apiserver, kube-scheduler, kube-controller-manager, kubelet, containerd/cri-o, CoreDNS, etcd, ingress-контроллеры, CNI, CSI

Системы интеграции: ETL пайплайны, Kafka, RabbitMQ, NGINX, Apache

CI/CD и инструменты поддержки: Harbor (целевой реестр), деплои, откаты, сбой дипломов и изменение конфигураций

4.6. Требования к хранению логов кластера и политике жизненного цикла

Для обеспечения возможности проведения пост-инцидентных расследований, аудита, анализа поведения систем, соответствия требованиям информационной безопасности и регуляторов, логи должны храниться в соответствии с гибкой политикой жизненного цикла, разделенной на горячее и холодное хранилище.

4.6.1. Характеристики для горячего хранения

- retention 30 дней;
- доступ через инструменты поиска как open source так как проприетарных решений;
- хранение логов обеспечить только на SSD носителях;
- обеспечить возможность fulltext search по любому выбранному индексу;

4.6.2. Характеристики холодного хранения

- retention до 6 месяцев (в зависимости от требования ЦБ Узбекистана);
- просмотр без возможности восстановления логов в горячее хранилище;
- данные необходимо хранить в S3 совместимом хранилище;

5. Требования к CI/CD

В целях обеспечения высокой скорости, устойчивости и управляемости поставки программного обеспечения и программных продуктов вендоров в архитектуре создаваемой системы должны быть реализованы и строго соблюдаться принципы непрерывной интеграции (CI) и непрерывной доставки (CD). CI/CD является неотъемлемой частью данного решения и обеспечивает фундамент для безопасной, контролируемой и воспроизводимой разработки и эксплуатации программных компонентов и систем.

Реализация CI/CD должна быть стандартизирована, унифицирована и масштабируема, с учетом особенностей микро-сервисной архитектуры и развертывания приложений в контейнерезированной среде Kubernetes.

5.1. Требования к системе контроля версии

Все манифесты, настройки пайплайнов и конфигураций должны храниться в централизированной системе контроля версий. Банком используется система Gitlab onprem.

Подрядчик должен использовать модель ветвления в основе которой лежит git-flow или использовать базовый git-flow.

5.2. Требования SAST/DAST процессам

Для каждого пайплайна необходимо реализовать возможность запуска инструментов SAST/DAST;

Все артефакты сборок должны храниться либо в Harbor, либо в Nexus;

PROD среда и размещение обновления в данной среде необходимо реализовать в ручную (без автоматизческой сбор и размещения после коммита изменений);

DEV/TEST/UAT необходимо полностью автоматизировать как rollout так и rollback;

Все пайплайны должны иметь возможность использовать встроенные механизм Vault для хранения секретов и паролей.

5.3. Установленный технологический стек

Категория	Инструменты
CI/CD инструменты, используемые в кластере	Gitlab CI/CD, Jenkins
SAST/DAST	SonarQube, Nessus
Реестр артефактов	Harbor, Nexus
Инфраструктурный код	Helm
Агенты	Только onprem агенты

6. Требования к доступности кластеров

Кластеры Kubernetes, на которых развернута система, должны быть спроектированы и реализованы таким образом, чтобы обеспечивать устойчивую, непрерывную и предсказуемую работу всех размещенных компонентов.

Для среды PROD/UAT

Масштабируемость	10+ node
Автоматизация поставок	Режим canary
HA	3+ master + 3+worker

DEV, TEST и UAT можно использовать минимально допустимую конфигурацию доступности.

6.1. Архитектурные требования обеспечения отказоустойчивости

Ниже приводится требования к среде PROD.

Требования для control plane:

- не менее трех управляющих узлов, работающих в режиме quorum;
- master ноды должны быть распределены по разным зонам доступности или разным физическим серверам;

Требования для балансировки нагрузки:

- использование балансировщика L4 уровня типа NGINX или HAProxy;
- ingress контроллер минимум в двух репликах;
- логирование failover ingress контроллера;

Требования для вычислительных нод:

- минимальный набор вычислительных нод 3;

- обязательное включение replicaSet для распределения нагрузки и синхронизации pod;

7. Требование к безопасности

Данный раздел описывает функциональные и организационно-технические меры, направленные на обеспечение комплексной информационной безопасности компонентов кластера, развернутых приложений, пользовательских данных и вспомогательной инфраструктуры.

- вход в кластер для его администрирования только через сертификаты или SSO;
- авторизация только через RBAC политики с выделенным ACL;
- запрещено использовать сервисные аккаунты;
- для всех namespace должны быть заданы сетевые правила, ограничивающие входящий и исходящий трафик между pod на на уровне сетевом и транспортном;
- в кластере должен быть активирован аудит Kubernetes (Audit Logs);
- audit logs должен быть передан в централизованное хранилище логов;
- audit logs должны быть переданы в SIEM IBM QRADAR;
- настроено оповещение по audit logs на почту ответственным лицам или же в систему мгновенных сообщений типа telegram;
- в рамках технической поддержки, поставщик услуги должен обеспечить минимальный SLA по обновлению версий компонентов платформы;
- соответствие требованию PCI-DSS и PA-DSS;
- на входящий трафик на ingress необходимо включить modsec;
- интеграция с DNS (внешним) для создания DNS записей;

8. Требование к лицензионной чистоте

- запрещено использовать компоненты, находящиеся под санкциями;
- запрещено использовать компоненты, находящиеся под лицензиями AGPL и SSPL и отдельных версий GNU GPL;
- все исходные коды пайплайнов, кода и баз данных, разработанных в рамках создания платформы, являются интеллектуальной собственностью банка;

9. Требования к документированию

В рамках реализации проекта, требуется полноценный набор документов по эксплуатации, мониторингу и разработки нового функционала в платформе. Необходимо обеспечить следующим набором документов в формате docx

Документ	Описание
Описание общей архитектуры системы (в т.ч. диаграммы компонентов, C4-модели, схемы взаимодействия).	Необходимо документировать архитектуру решения до уровня Level3 в нотации C4
Список принятых ADR по платформе	Необходимо задокументировать все принятые архитектурные решения при разработки платформы.

Руководства по установке, запуску, остановке, масштабированию компонентов.	В данную документацию так же должна входить инструкция по развертыванию нового кластера с нуля
Инструкции работы с системой мониторинга	
Инструкция работы с логирование	
Структура и описание исходного кода, API-эндпоинтов, схемы БД.	Все написанные кастомные компоненты, базы данных и API необходимо описать в документации
Документ описывающий все используемые форматы обмена данными (kafka, AMQP и т.д.). Если будут использоваться на системном уровне	В данной документации необходимо описать все JSON или форматы обмена данными, описать каждый атрибут
Регламенты и политики безопасности (например, RBAC, NetworkPolicy, TLS-цепочки).	Какие приняты регламенты безопасности в платформе. Как мы используем RBAC и так далее
Руководства для операторов, разработчиков, аналитиков, тестировщиков.	Для каждой роли следует описать инструкции работы с платформой
Интерфейсы для взаимодействия с системой (CLI, WebUI, дашборды).	Для каждого интерфейса взаимодействия с платформой, необходима подробная инструкция работы с ней

10. Ограничения и допущения

Ограничения:

- в рамках разработки платформы, не подразумевается портирование текущих банковских систем силами выбранного подрядчика;
- использование исключительно внутренней (on-premise) инфраструктуры для PROD-сред, без размещения критических компонентов в публичных облаках.
- операционные системы и ПО должны соответствовать утверждённому списку сертифицированных и поддерживаемых версий (oracle linux, oracle database, Postgresql 12+ и т.д.)
- интеграция с внешними системами возможна только через REST, json-rpc
- срок хранения логов и бэкапов должен соответствовать требованиям внутреннего ИБ-политики (не менее 6 месяцев).
- все изменения, затрагивающие продуктивную инфраструктуру, должны проходить через RFC-процесс в рамках ITSM
- поддержка только одной выбранной системы CI/CD на весь стек

Допущения:

- предполагается наличие выделенной команды DevOps-инженеров и специалистов по сопровождению Kubernetes для поддержки инфраструктуры продолжительностью не менее 4х месяцев;
- Считается, что внутри организации уже действует или будет внедрена система централизованного логирования (например, ELK, OpenSearch);
- банк предоставляет доступ к инфраструктуре, сетевым ресурсам, доменам, DNS, и внутренним реестрам в рамках утвержденных сроков;
- все взаимодействующие команды будут придерживаться установленных самостоятельно стандартов ведения документации и версионирования артефактов (semver);
- подразумевается наличие внутренней службы ИБ, которая будет выполнять аудит, согласование политик и ревизию RBAC (внутренний аудит);
- обновления ПО будут выполняться согласно план-графику, в выделенные временные окна, согласованные с ИТ владельцем платформы;

Boshqaruv Raisi o'rinbosari: **D.Umarov**

kelishuvchilar: V.Krasnov, A.Kenjayev, A.Ergashev

https://hujjat.sqb.uz/?pin=aP52vU38&id=782bccf6-2cd8-48ff-8496-8f43986599f9