

АО «Финтек Системс»

КОМПЬЮТЕРНАЯ ПРОГРАММА (ПО) ДЛЯ ЭВМ «ИСТОК»
Документация, необходимая для установки программного обеспечения

2025 г.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ	4
1.1. Назначение программного комплекса	4
1.2. Область применения программного обеспечения	4
1.3. Функциональные характеристики	4
2. СОСТАВ РАЗВЕРТЫВАНИЯ	6
3. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	7
3.1. Общие сведения об администрировании	7
3.2. Управление лицензиями	7
3.3. Установка FluxCD (bootstrap)	8
3.4. Установка Sealed-secrets	9
3.5. Установка PostgreSQL Server	10
3.6. Установка MongoDB	14
3.7. Сборка приложения	15
3.8. Установка ПО ИСТОК	16
3.9. Проверка работы	18
4. КОНТАКТНЫЕ ДАННЫЕ СЛУЖБЫ ПОДДЕРЖКИ	18

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

1.1. Назначение программного комплекса

Программное обеспечение для ЭВМ предназначена для обработки пользовательских действий, связанных с интерактивными ставками на спортивные события, а также для выполнения операций, инициируемых по результатам наступления соответствующих событий.

Система реализована в виде веб-ориентированного программного комплекса с доступом через стандартные веб-браузеры на настольных и мобильных устройствах.

Исключительные права на текущую версию решения принадлежат АО «Финтек Системс».

1.2.Область применения программного обеспечения

Программное обеспечение для решения отраслевых задач в области культуры, спорта, организации досуга и развлечений Программное обеспечение, предназначенное для специализированного информационного обеспечения в области творческой деятельности, деятельности в области искусства, культуры, спорта, организации досуга и развлечений, деятельности библиотек, архивов, музеев и прочих объектов культуры, деятельности по организации и проведению азартных игр и заключению пари, по организации и проведению лотерей.

1.3.Функциональные характеристики

Программное обеспечение используется для:

- учета и обработки информации об интерактивных ставках;
- хранения и актуализации структурированных данных;
- контроля ограничений и отказов в приеме интерактивных ставок;
- привязки интерактивных ставок к спортивным событиям;
- обработки результатов спортивных событий;
- определения итогов пари и расчета выигрышей;

- учета выплаченных и невыплаченных выигрышей;
- формирования отчетных и производных показателей;
- интеграции с внешними информационными системами и информационными системами государственных регуляторов;
- обеспечения соответствия требованиям законодательства и информационной безопасности.

2. СОСТАВ РАЗВЕРТЫВАНИЯ

2.1.Основные положения

2.1.1. Компьютерная программа (ПО) для ЭВМ «ИСТОК» реализовано посредством микросервисной архитектуры.

2.1.2. Клиент – любая операционная система, имеющая соединение с Интернетом и способная отображать web-страницу или мобильное устройства типа смартфона.

2.1.3. Требования к серверу:

- Процессор: 8 ядер, 3GHz и выше
- Оперативная память (RAM): 60 Gb и более
- Система хранения данных (HDD/SSD): от 100Gb, рекомендуется RAID 1, не менее 3000 IOPS

2.1.4. Требования к ПО:

- Kubernetes 1.33.7, отказоустойчивый (3 master ноды)
- FluxCD v2.4.0
- Postgres 16.2.2
- Mongodb 14.3.2

Должны быть установлены все актуальные обновления для используемой версии операционной системы.

2.1.5. Развёртывание в продуктивной среде производится посредством публикации программного обеспечения после успешного прохождения интеграционного и регрессионного тестирования.

Язык: Node.js, JavaScript, HTML и CSS.

3. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1.Общие сведения об администрировании

3.1.1. Администратор обеспечивает надлежащую работоспособность платформы, выполняя функции администрирования платформы. К основным функциями администратора платформы относятся:

- Установка и сопровождение ПО: установка и настройка ОС, прикладного ПО на серверах приложения, СУБД, рабочих местах пользователей, мониторинг работоспособности ПО.
- Обеспечение сохранности и актуальности данных: создание и поддержание необходимого количества резервных копий, восстановление данных при необходимости.
- Управление учетными записями пользователей: регистрация пользователей, контроль использования учетных записей, задание ограничений на использование учетных записей, блокировка и удаление неиспользуемых учетных записей.
- Защита данных от несанкционированного доступа: настройка учетных записей пользователей путем установки соответствующих прав пользователей.

3.1.2. В своей работе администратор платформы должен руководствоваться следующими документами:

- Описание функциональных характеристик;
- Технической документацией по ОС

3.2.Управление лицензиями

3.2.1. Для использования ПО не предусмотрены лицензионные ключи. ПО разворачивается и настраивается индивидуально в инфраструктуре заказчика. АО “ФИНТЕК СИСТЕМС”, обладает исключительным правом на регистрируемую компьютерную программу (ПО) для ЭВМ “ИСТОК”.

3.3. Установка FluxCD (bootstrap)

Рис. 1. Настройки fluxcd

```
flux bootstrap gitlab \
--owner=dev1693130 \
--repository=fluxcd \
--branch=main \
--path=clusters/selectel-moscow-ru-7 \
--deploy-token-auth
```

Далее проверяем наличие всех установленных компонентов:

Рис. 2. обзор компонентов fluxcd

k get pods -n flux-system				
NAME	READY	STATUS	RESTARTS	AGE
helm-controller-76bd84c65d-gs8nl	1/1	Running	0	25h
kustomize-controller-766fb88648-6xl2k	1/1	Running	0	25h
notification-controller-7d6f99878b-mm69t	1/1	Running	0	25h
source-controller-66bb88f74f-dt2k6	1/1	Running	0	25h

Ключевые особенности конфигурации

- **source-controller** - забирает код и чарты из Git/Helm/OCI-репозиториев и отслеживает их изменения.
- **kustomize-controller** - применяет YAML-манифесты из Git в кластер и следит за их соответствием.
- **helm-controller** - устанавливает и обновляет Helm-чарты через HelmRelease и контролирует их состояние.
- **notification-controller** — отправляет уведомления о событиях и статусах Flux во внешние системы.

3.4. Установка Sealed-secrets

Sealed Secrets позволяет хранить Kubernetes Secret в Git в зашифрованном виде.

Расшифровка происходит только внутри кластера специальным контроллером.

Рис. 3. Настройки sealed secrets

```
helm repo add sealed-secrets https://bitnami-labs.github.io/sealed-secrets
helm repo update
helm install sealed-secrets \
  sealed-secrets/sealed-secrets \
  -n sealed-secrets \
  --create-namespace
```

Проверяем появление сервиса:

Рис. 4. проверка sealed secrets

```
k get pods -n sealed-secrets
NAME                  READY   STATUS    RESTARTS   AGE
sealed-secrets-6d7644c759-9fthg   1/1     Running   0          1d
```

3.5. Установка PostgreSQL Server

Типовая конфигурация выглядит следующим образом. Разворачиваем ее с помощью fluxcd

Рис. 5. конфигурация postgres

```
---  
apiVersion: helm.toolkit.fluxcd.io/v2beta2  
kind: HelmRelease  
metadata:  
  name: postgres17-svc-pam  
  namespace: istok-infra  
spec:  
  chart:  
    spec:  
      chart: postgresql  
      reconcileStrategy: ChartVersion  
      sourceRef:  
        kind: HelmRepository  
        namespace: flux-system  
        name: bitnami  
        version: 16.2.2  
  install:  
    createNamespace: true  
  interval: 10m0s  
  targetNamespace: istok-infra  
values:  
  global:  
    storageClass: fast.ru-7a  
  postgresql:  
    auth:  
      database: pam  
      username: pam  
      existingSecret: postgres17-svc-pam-auth  
      secretKeys:  
        adminPasswordKey: "postgres-password"  
        userPasswordKey: "password"  
        replicationPasswordKey: "replication-password"  
    fullnameOverride: postgres17-svc-pam  
    postgresqlSharedPreloadLibraries: "pgaudit, pg_stat_statements"  
    architecture: replication  
    primary:  
      persistence:  
        size: 20Gi  
      resources:  
        requests:  
          cpu: 50m  
          memory: 300Mi  
        limits:  
          cpu: 1000m  
          memory: 1500Mi  
      extendedConfiguration: |  
        wal_level = logical  
        autovacuum_vacuum_scale_factor = 0.1  
        autovacuum_vacuum_threshold = 200  
        effective_cache_size = 1GB  
        maintenance_work_mem = 1GB  
        max_connections = 200  
        max_wal_size = 2GB  
        shared_buffers = 2GB  
        wal_buffers = 32MB  
        wal_keep_size = 10GB  
        work_mem = 1MB  
        log_min_duration_statement = 250  
        log_line_prefix = '%m [%p] %q%u@%d '
```

Ключевые особенности конфигурации

1. Проверенный источник и версия

Используется Helm-чарт postgresql версии 16.2.2 из официального репозитория Bitnami, обеспечивающий стабильность и предсказуемость обновлений.

2. Безопасное управление доступом

Все учетные данные PostgreSQL хранятся во внешнем Kubernetes Secrets-сервисе, что упрощает ротацию паролей и соответствует требованиям безопасности.

3. Выделенное и производительное хранилище

Для хранения данных используется PVC объёмом 20 Gi на выделенном StorageClass fast.ru-7a, оптимизированном под I/O-нагрузку.

4. Контролируемое потребление ресурсов

Для primary-узла заданы явные CPU и memory requests/limits, что обеспечивает баланс между эффективной резервацией ресурсов и защитой от перегрузки.

5. Production-настройки PostgreSQL

Включены расширения (pgaudit, pg_stat_statements) и заданы параметры autovacuum, памяти и логирования медленных запросов для эксплуатации в продакшене.

Для каждого микросервиса настраиваем отдельный PostgreSQL.

Для шифрования паролей используем следующий шаблон:

Рис. 6. конфигурация для шифрования

```
---  
apiVersion: v1  
kind: Secret  
metadata:  
  name: postgres17-svc-pam-auth  
  namespace: istok-infra  
  type: Opaque  
stringData:  
  password: some_password  
  postgres-password: some_password
```

Генерируем надежные пароли, минимум в 30 знаков разного регистра, с добавлением спецсимволов.

Для дальнейшего безопасного хранения конфигурации, необходимо зашифровать конфигурацию следующий командой:

Рис. 7. шифрование паролей

```
kubeseal \
--controller-name=sealed-secrets \
--controller-namespace=sealed-secrets \
--allow-empty-data \
-f postgres17-svc-pam-auth.secret.yaml -o yaml > postgres17-svc-pam-auth.secret.sealed.yaml
```

3.6. Установка MongoDB

Типовая конфигурация выглядит следующим образом. Разворачиваем ее с помощью fluxcd:

Рис. 8. конфигурация mongodb

```
---
```

```
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: contentblender-mongodb-istok
  namespace: istok-infra
spec:
  chart:
    spec:
      chart: mongodb
      reconcileStrategy: ChartVersion
      sourceRef:
        kind: HelmRepository
        namespace: flux-system
        name: bitnami
        version: 14.3.2
  install:
    createNamespace: true
  interval: 10m0s
  targetNamespace: istok-infra
  values:
    image:
      registry: public.ecr.aws
      repository: bitnami/mongodb
      tag: "7.0.4-debian-11-r4"
      pullPolicy: IfNotPresent
      fullnameOverride: contentblender-mongodb
      architecture: standalone
      useStatefulSet: true
    auth:
      enabled: true
      rootUser: root
      existingSecret: contentblender-mongodb
      # passwords in existing secret
      usernames:
        - contentblender
    databases:
      - contentblender
  resources:
    requests:
      cpu: 100m
      memory: 500Mi
    limits:
      cpu: 3000m
      memory: 6500Mi
  persistence:
    enabled: true
    storageClass: fast.ru-7a
    size: 20Gi
  backup:
    enabled: false
  arbiter:
    enabled: false
  metrics:
    enabled: false
```

3.7. Сборка приложения

Собираем образ приложения и отправляем его в реестр:

Рис. 9. сборка образов

```
docker build -t istokregistry/pam:2025.12.26 -f .deploy/pam.Dockerfile .
docker push istokregistry/pam:2025.12.26
```

Инструкции для сборки образа выглядит следующим образом:

Рис. 10. dockerfile

```
FROM node:22.17.0-slim AS base

RUN apt-get update -y && apt-get install -y openssl
FROM base AS builder

WORKDIR /app

COPY . .

RUN npx turbo@2.4.0 prune --scope=pam --docker

FROM base AS installer

WORKDIR /app

COPY --from=builder /app/out/json/ .

RUN corepack enable && yarn install

COPY --from=builder /app/out/full/ .

RUN yarn proto:generate
RUN yarn prisma:generate

RUN yarn build --filter=pam...

FROM base AS runner

ENV PORT=3000

WORKDIR /app

COPY --from=installer /app/services/pam/build .

RUN addgroup --system --gid 1001 nodejs
RUN adduser --system --uid 1001 appjs

RUN chown -R appjs:nodejs /app
RUN npm install -g prisma@5.12.1
USER appjs

EXPOSE 3000

CMD ["node", "index.js"]

ARG IMAGE_TAG=none
ENV IMAGE_TAG=${IMAGE_TAG}
```

3.8. Установка ПО ИСТОК

Для установки приложения необходимо передать конфигурацию во fluxcd. Типовой манифест выглядит следующим образом:

Рис. 11. конфигурация приложения

```
---  
apiVersion: helm.toolkit.fluxcd.io/v2beta1  
kind: HelmRelease  
metadata:  
  name: istok-svc-pam-release-istok  
  namespace: istok  
spec:  
  chart:  
    spec:  
      chart: ./deploy/helm/svc-pam  
      reconcileStrategy: ChartVersion  
      sourceRef:  
        kind: GitRepository  
        namespace: flux-system  
        name: deploy  
        valuesFiles:  
          - ./deploy/helm/svc-pam/values.yaml  
      interval: 1m0s  
    releaseName: istok-svc-pam-release-istok  
    targetNamespace: istok  
    values:  
      envName: istok  
      fullnameOverride: istok-svc-pam  
      replicaCount: 1  
      image:  
        tag: istok-2025.12.26.1  
        pullPolicy: IfNotPresent  
      resources:  
        requests:  
          cpu: 100m  
          memory: 300Mi  
        limits:  
          cpu: 300m  
          memory: 1500Mi  
      startupProbe:  
        httpGet:  
          path: /k8s/startup  
          port: http  
        periodSeconds: 10  
        timeoutSeconds: 10  
        failureThreshold: 18  
      livenessProbe:  
        httpGet:  
          path: /k8s/liveness  
          port: http  
        periodSeconds: 20  
        timeoutSeconds: 10  
        failureThreshold: 3  
      readinessProbe:  
        httpGet:  
          path: /k8s/readiness  
          port: http  
        periodSeconds: 5  
        timeoutSeconds: 5  
        failureThreshold: 2
```

3.9. Проверка работы

Необходимо получить список действующий сервисов. Миграции должны быть в статусе **Completed**, а контейнеры приложения в статусе **Running**

Проверяется наличие ресурсов в Ingress и доступность сервиса при обращении к приложению через веб-браузер

Рис. 12. список ingress

		CLASS	HOSTS	ADDRESS	PORTS	AGE
app-contentblender	nginx	contentblender.istok.ftsystems.ru,backoffice.istok.ftsystems.ru		155.212.180.176	80, 443	3h10m
app-main	nginx	istok.ftsystems.ru		155.212.180.176	80, 443	3h10m
svc-big	nginx	api.istok.ftsystems.ru,api-v2.istok.ftsystems.ru		155.212.180.176	80, 443	3h9m
svc-kyc	nginx	kyc.istok.ftsystems.ru		155.212.180.176	80, 443	3h19m
svc-payment-gateway	nginx	payment-gateway.istok.ftsystems.ru		155.212.180.176	80, 443	3h22m
svc-sportsbook-phoenix	nginx	sport.b2b.istok.ftsystems.ru		155.212.180.176	80, 443	3h19m
svc-traffic	nginx	traffic.istok.ftsystems.ru		155.212.180.176	80, 443	3h19m

4. КОНТАКТНЫЕ ДАННЫЕ СЛУЖБЫ ПОДДЕРЖКИ

Адрес портала службы поддержки: sd@ftsystems.ru

Тел.: +7 (800) 700-62-08

Электронная почта: info@ftsystems.ru