

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ**

ПО E-Voting

Москва, 2026

Оглавление

Общие сведения	3
Обозначение и наименование Программы.....	3
Функциональные характеристики.....	3
Назначение Программы	3
Функциональные возможности программы.....	3
2 УСТАНОВКА СИСТЕМЫ	6
2.1 Состав пакетов	6
2.2 Основные контуры.....	7
2.2.1 Операционная система.....	8
2.2.2 Аппаратные характеристики	8
2.3 Архитектура системы	9
2.4. Основные модули и их назначение.....	10
2.4 Конфигурация окружения.....	11
2.4.1 Переменные окружения tomcat	11
2.4.2 Настройки ActiveMQ Artemis	19
2.4.3 Настройки БД Postgres	19
2.4.4 Настройки проксирования nginx	20
2.4.5 Настройки redis	22
2.5 Порядок установки	22
2.5.1 Установка базового ПО	22
2.5.2 Развертывание BackEnd	31
2.5.3 Развертывание FrontEnd	31
2.5.4 Проверка работоспособности после установки	32
2.5.5 Удаление/откат	32
3. Перечень терминов и сокращений	33

ОБЩИЕ СВЕДЕНИЯ

Обозначение и наименование Программы

Полное наименование: ПО E-voting.

Условные обозначения: E-voting, Система.

ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ

Назначение Программы

ПО E-voting предназначена для создания портала электронного голосования и реализации комплекса мер с целью обеспечения возможности дистанционного участия на заседаниях, обсуждения вопросов повестки дня и принятия решений по вопросам, поставленным на голосование, без присутствия в месте проведения заседания владельцев ценных бумаг и членов коллегиальных органов российских компаний в рамках исполнения требований законодательства по проведению общих собраний акционеров, владельцев облигаций, и иных ценных бумаг, а также заседаний других коллегиальных органов управления российских компаний, на которых предусматривается заполнение электронной формы бюллетеня при проведении заочных голосований и дистанционных заседаний клиентов НКО АО НРД («НРД»).

E-voting предназначена для автоматизации процессов организации и проведения общего собрания владельцев ЦБ и коллегиальных органов (далее – участник), проводимого в форме заседания и (или) заочного голосования, включая:

- создание, корректировку, активацию и отмену собрания;
- выгрузку из E-voting информации о собрании;
- загрузку в E-voting информации о собрании, материалов и электронной формы бюллетеня для голосования;
- загрузку в E-voting списка участников собрания (списка лиц, имеющих право голоса при принятии решений общим собранием);
- предоставление доступа к собранию участникам, наблюдателям, представителям эмитентов, регистраторов, организаторам собрания, членам счетной комиссии;
- голосование в электронном виде, включая регистрацию на заседание с дистанционным участием;
- осуществление информационного сопровождения по ходу проведения собрания;
- формирование отчетов после окончания собрания и в процессе его проведения;
- предоставление доступа к архивной информации по прошедшим собраниям.

Функциональные возможности программы

ПО E-voting реализует следующие основные прикладные функции:

- авторизацию пользователей и единую точку доступа пользователей к системе;
- получение участникам перечня доступных собраний;
- получение участниками и наблюдателями материалов собрания;
- получение участниками и наблюдателями информационных сообщений;
- предоставление доступа пользователей к аудио и видеотрансляции;

- заполнение электронной формы бюллетеня;
- сохранение заполненной электронной формы бюллетеня;
- регистрацию на заседание с дистанционным участием;
- отправку электронных бюллетеней в Счетную комиссию;
- выгрузку владельцами ЦБ:

персонифицированного бюллетеня для голосования с заполненными результатами голосования (содержание электронной формы бюллетеня – для владельца ЦБ, документ о голосовании – для иных участников голосования);

форма бюллетеня для голосования, применявшегося на собрании (информация из бюллетеня для голосования – для владельцев ЦБ, форма документа о голосовании – для иных участников голосования);

выписки о регистрации для участника заседания;

выписки о голосовании по вопросам повестки дня.

- регистрацию (инициацию) собрания;
- ввод информации о собрании;
- сбор списков участников собрания;
- загрузку и размещение на портале материалов собрания;
- загрузку и размещение на портале бюллетеня для голосования на собрании;
- управление параметрами;
- брендирование интерфейса для проведения собрания и голосования;
- управление составом наблюдателей и их правами доступа в рамках собрания;
- управление SMS и e-mail уведомлениями;
- размещение информационных сообщений для участников собрания;
- предоставление ответов на вопросы участников, включая на запросы выступления на дистанционном заседании;
- загрузку внешних источников данных о голосовании;
- актуализацию и предоставление пользователям информации о статусе собрания;
- ручной ввода итогов голосования в web-кабинете счетной комиссии;
- формирование и выгрузка протокола по результатам голосования из внешних источников;
- загрузку из внешних источников протокола по результатам голосования;
- формирование и выгрузка отчета по результатам голосования;
- размещение отчета по результатам голосования;
- формирование отчетов по активным собраниям:

статистика голосования в разрезе вопросов повестки дня;

форма бюллетеня для голосования, применявшегося на собрании;

протокол общего собрания;

протокол об итогах голосования;

отчет об итогах голосования;

информация о кворуме;

- отправку собрания в архив;
- резервное копирование информации активных собраний.
- хранение информации по прошедшим собраниям;

- формирование отчетов по прошедшим собраниям:
форма бюллетеня для голосования, применявшегося на собрании;
протокол общего собрания;
протокол об итогах голосования на собрании;
отчет по результатам голосования.

Формы инициации собрания обеспечивают возможности:

- создания нового собрания (мероприятия);
- ввода общей информации по собранию;
- сохранения введенной информации по собранию;
- просмотра информации по собранию.

Формы результатов голосования обеспечивают возможности:

- просмотра информации о ходе голосования;
- подготовки протокола;
- подготовки отчета по результатам голосования и других отчетов, формируемых компонентом;
- просмотра протокола;
- просмотра отчета по результатам голосования.

Формы администрирования обеспечивают возможности:

- управления параметрами проведения собрания;
- управления составом наблюдателей;
- управления полномочиями наблюдателей;
- публикации информационных сообщений для участников собрания.

Формы загрузки и выгрузки информации обеспечивают:

- инициацию загрузки информации из системы;
- инициацию загрузки в систему информации из внешних источников;
- отображение хода загрузки информации;
- просмотр и сохранение информации, загруженной из внешнего источника.

Формы заполнения бюллетеня обеспечивают возможности:

- просмотра бюллетеня;
- просмотра информации о количестве ценных бумаг по состоянию на дату фиксации списка лиц, имеющих право голоса при принятии решений общим собранием участников;
- заполнения бюллетеня;
- сохранения заполненного бюллетеня;
- просмотра перечня заполненных бюллетеней;
- просмотра и корректировки сохраненного бюллетеня;
- инициации отправки заполненного бюллетеня;
- формирования и отправки заявки для участия в проведении общего собрания владельцев ценных бумаг;
- формирования просмотра и выгрузки персонифицированного бюллетеня и выписок.

Формы участника собрания обеспечивают возможности:

- выбора общего собрания из перечня доступных участнику;

- просмотра и скачивания материалов по собранию;
- просмотра информационных сообщений Организатора собрания и Счетной комиссии;
- отправку вопросов, включая запросы выступления на заседании с дистанционным участием, и просмотр ответов на них;
- просмотра отчета по результатам голосования.

2 УСТАНОВКА СИСТЕМЫ

2.1 Состав пакетов

Таблица 1 – Состав пакетов установки

Пакет	Контур	Назначение
idp-core.war	SRV	Ядро модуля idp
idp-web.war	WEB	Веб-интерфейс пользователя авторизации
idp-app.js	Nginx	Фронтенд Angular
portal-bkg.war	WRK	Функционал исполнения тасок
portal-core.war	SRV	Ядро модуля portal
portal-emulation.war	SRV	Sms/email
portal-interop.war	SRV	Интеграции регистратор и пр.
portal-web.war	WEB	Веб-интерфейс администратора
public-bkg.war	WRK	Функционал исполнения тасок
public-core.war	SRV	Ядро модуля public

public-web.war	WEB	Веб-интерфейс пользователя участника собраний
public app js	Nginx	Фронтенд Angular
archive-bkg.war	CH	Функционал исполнения тасок
archive-core.war	CH	Ядро модуля archive
archive-web.war	WEB	Веб-интерфейс пользователя архива
event-logging.war	CH	Модуль логирования антифрод событий
business-logging.war	CH	Модуль логирования бизнес событий
common-report.war	WRK	Модуль отчетов

2.2 Основные контуры

Таблица 2 – Описание контуров установки

Наименование	Сервера	Назначение
WEB – контур tomcat	p02web01.ndcw.ru, p02web02.ndcw.ru	Функционал web интерфейса и модуля авторизации
SRV - контур tomcat	p02srv01.ndcw.ru, p02srv02.ndcw.ru	Функционал core части основных модулей
WRK – контур tomcat	p02wrk01.ndcw.ru, p02wrk02.ndcw.ru	Функционал работы bkg модулей

CH - контур tomcat	p02ch01.ndcw.ru, p02ch02.ndcw.ru	Функционал архива и логов
Nginx	p02nlb01.ndcw.ru, p02nlb02.ndcw.ru	Веб – сервер для фронтенда Angular
ActiveMQ Artemis	p02qr01.ndcw.ru, p02qr02.ndcw.ru	Брокер сообщений

2.2.1 Операционная система

На всех серверах всех контуров установлена операционная система RedOS 8

2.2.2 Аппаратные характеристики

Аппаратные характеристики серверных компонентов указаны в документе «Описание используемого технологического стека ПО E-Voting» раздел «Технические средства»

2.3 Архитектура системы

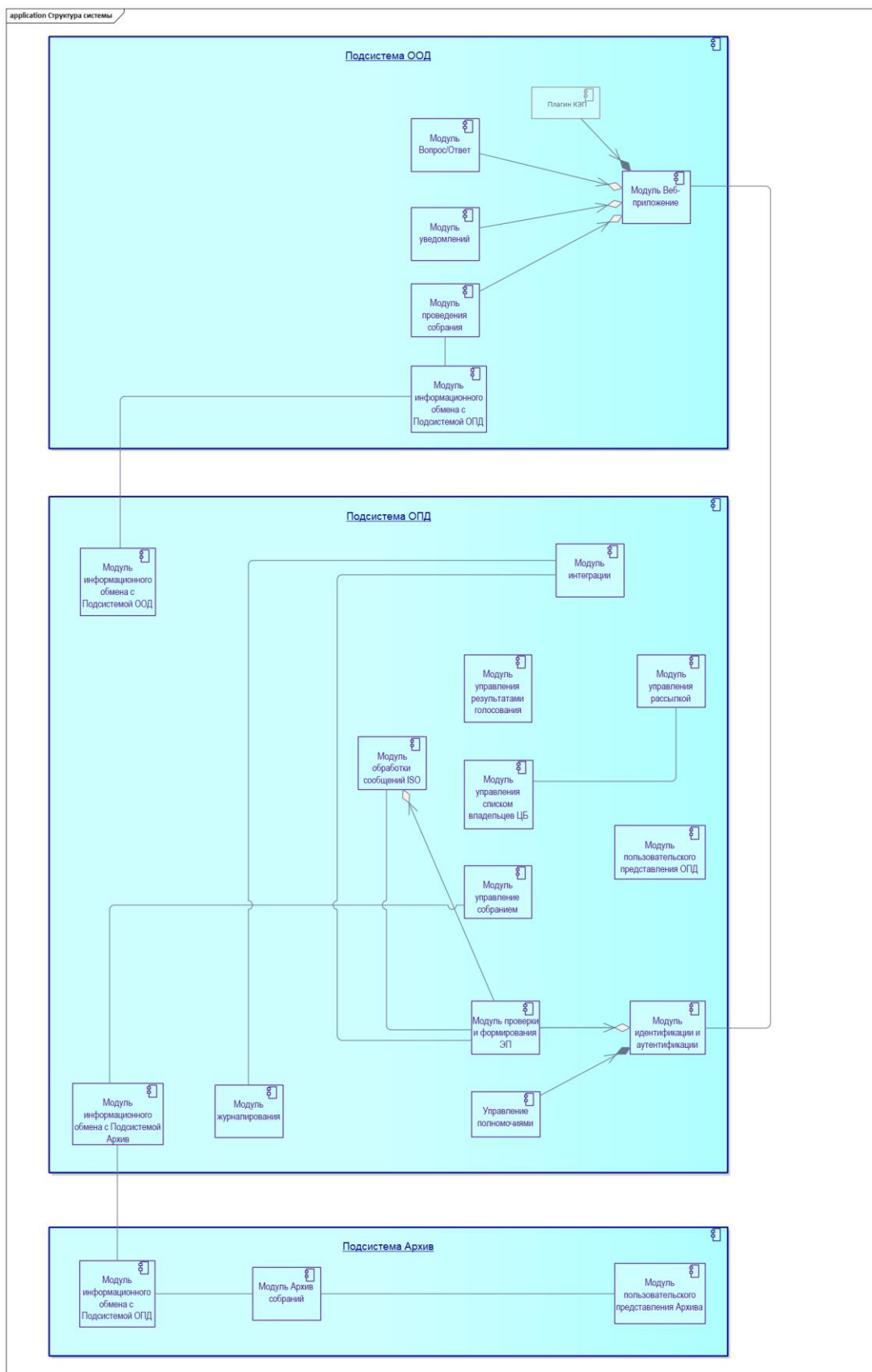


Рисунок 1. Структурная схема системы E-voting

Проект представляет собой модульный монолит на платформе Java EE.

Для функционирования ПО E-voting на серверах контура используется следующее базовое программное обеспечение:

- среда виртуализации — Kubernetes;
- веб-сервер / reverse proxy — Nginx;
- сервер приложений — Apache Tomcat;
- СУБД — PostgreSQL;
- брокер сообщений — Apache ActiveMQ Artemis Message Broker;
- in-memory хранилище — Redis;
- управление кластером высокой доступности — Pacemaker;
- криптопровайдер — КриптоПро CSP;
- средство управления конфигурацией — Ansible;
- система мониторинга — Zabbix.

2.4. Основные модули и их назначение

Таблица 3 – Описание модулей

Наименование	Назначение
IDP	Модуль авторизации, использует интеграции с ЕСИА и сбер ID
Portal	Модуль создания и настройки собраний
Public	Модуль проведения собраний
Archive	Модуль архива собраний
Business-logging	Модуль логирования бизнес событий
Event-logging	Модуль логирования событий для антифрода и iasgop

Report	Модуль формирования отчетов
--------	-----------------------------

Модули для взаимодействия друг с другом используют как синхронные сетевые вызовы с помощью протокола thrift, так и асинхронные взаимодействия с помощью брокера ActiveMQ Artemis.

Веб интерфейс реализован на Angular версии 2.4

Используется БД Postgres

2.4 Конфигурация окружения

2.4.1 Переменные окружения tomcat

В таблице 4 приведены переменные окружения, задаваемые для сервера приложений Apache Tomcat. Переменные определяют параметры подключения к сервисам аутентификации, интеграциям, пути к ресурсам, настройки безопасности и параметры обмена сообщениями.

Таблица 4 – Переменные окружения

Переменная	Описание	Значение для продуктового сервера
idp.core.path	Idp path	http://evoting-svc.ndcw.ru/idp-core
portal.core.path	Portal meeting service url	http://evoting-svc.ndcw.ru/portal-core
idp.default.redirect	Адрес для перенаправления по умолчанию	https://online.e-vote.ru/e-voting-ui/public
idp.profile.url	Адрес профиля	https://online.e-vote.ru/e-voting-ui/idp/profile
sms.block.delay	Период на который нужно блокировать	60

	отправку СМС после отправки одной. В секундах	
sms.validity.delay	Период в течении которого СМС с кодом подтверждения действительна. В секундах	86400
idp.cookie.domain	Authentication token cookie domain	.e-vote.ru
idp.cookie.path	Authentication token cookie path descriptor	/
idp.cookie.isSecure	Authentication token cookie secure descriptor	false
idp.cookie.httpOnly	Authentication token cookie http only descriptor	true
recaptcha.url		http://ibmdp.dmzp.local:8001/recaptcha/api/siteverify

cache.expires.postage	Время хранения	31
count.attempts	Количество попыток	10
smev.service.adapter	Smev url	http://smev-evt-v001p.ndcw.ru:7575/ws/
smev.is.id	Smev id	852901_3S
smev.files.path	Smev path	/u01/smev-files/
smev.task.check.enable	Smev check	false
idp.esia.auth.url	ESIA auth url	https://esia.gosuslugi.ru/aas/oauth2/ac
idp.esia.clientId	ESIA client id	852902
idp.esia.scope.org	ESIA user org scope	http://esia.gosuslugi.ru/org\inf?org\oid={0}
idp.esia.redirectTo	ESIA redirect to process url	https://online.e-vote.ru/e-voting-ui/idp/sso/rs/esia/process
idp.esia.access.token.url	Access token	http://ibmdp.dmzp.local:8001/aas/oauth2/te
idp.esia.rest.url	ESIA rest endpoint path	http://ibmdp.dmzp.local:8001/rs

idp.esia.slo	ESIA logout url	https://esia.gosuslugi.ru/idp/ext/Logout?client\id=852902&redirect\url=https://online.e-vote.ru/e-voting-ui/public
ipd.esia.cert.name		t3crt
moex.module.name	IDP MOEX module name	evoting
moex.endpoint.url	IDP MOEX endpoint url	http://proxyp.nsd.ru/passport-moex/authenticate
idp.slo.url	Logout request	https://online.e-vote.ru/e-voting-ui/idp/slo/?redirectTo=https://online.e-vote.ru/e-voting-ui/portal/
public.url	Адрес публика (используется для перенаправления пользователей)	https://online.e-vote.ru/e-voting-ui/public/
message.url		https://online.e-vote.ru
archive.url	Адрес архива	https://online.e-vote.ru/e-voting-ui/archive
idp.path	Адрес публичног о сервиса IDP для обновления токена и получения	http://evoting.nsd.ru/idp

	списка плагинОВ	
public.core.path	Public sync service	http://evoting-svc.ndcw.ru/public-core
report.path	Report base path	http://int.evoting.nsd.ru/report_portal
check.intermediate.certificates.revocation		false
portal.core.keystore.path	сертификаты	~/keystore/prod/cacerts.jks
meeting.message.misfire.threshold		180
meeting.message.misfire.threshold.percent		20
allowed.file.types	Перечень доступных к загрузке файлов, через запятую	doc
cache.expire.default		300
password.validity.period		90
password.start.period		-14
password.end.period		10

smb.username	Samba имя пользователя	Evoting
smb.host	Samba адрес	fsoffice.nsd.ru
smb.share	Samba каталог	Depo
smb.path	Samba путь	OCBO
smb.domain	Samba домен	nsd.ru
meeting.instruction.load.new	Путь для не обработанных файлов для загрузки МІ для ВМЕТ	new
meeting.instruction.load.old	Путь для успешно обработанных файлов для загрузки МІ для ВМЕТ	old
meeting.instruction.load.error	Путь для обработанных с ошибкой файлов для загрузки	error

	МІ для ВМЕТ	
meeting.instruction.load.error.duplicate	Путь для обработанных с ошибкой - дубль файлов для загрузки МІ для ВМЕТ	duplicate
meeting.instruction.load.error.investigation	Путь для обработанных с ошибкой файлов для загрузки МІ для ВМЕТ	investigation
meeting.instruction.max.files	Максимальное количество файлов для одной обработки загрузки МІ для ВМЕТ	1000
scheduler.delay	Таймаут для селекта свободного промокода	3
nsd.service.file.storage.url	Content Manager url	http://file-storage.s-kubepr.ndcw.ru

weblogic.cf	Weblogic factory	jndi/mcvCF
weblogic.destination	Weblogic входящая очередь МПГ	jndi/meeting-instruction-evtn-g-mpg-event
weblogic.consumer.destination	Weblogic исходящая очередь МПГ	jndi/meeting-instruction-mpg-evtn-g-event
weblogic.provider.url	url weblogic	t3://mpg-app-mpgserver.ndcw.ru:7003
weblogic.security.principal	НУЗ	ta\w\levote_prod
report.template.path	Путь к шаблонам отчетов	~/NSD_HOME/portal/report/templates
report.locale.path	Путь к настройкам отчетов	~/../common/WEB-INF/locale
meeting.url		http://evoting.nsd.ru/public/#/meeting/{0}
idp.sso.url	Адрес для логина пользователя	https://online.e-vote.ru/e-voting-ui/idp/sso/?redirectTo=https://online.e-vote.ru/e-voting-ui/portal&client=PORTAL
redis.master		Mymaster
redis.sentinel.1.node		redis-sentinel:26379

mts.link.register.url	Адрес для регистрации и на mts link	https://userapi.mts-link.ru/v3/eventsessions/{sessionId}/register
-----------------------	-------------------------------------	---

2.4.2 Настройки ActiveMQ Artemis

Для обеспечения асинхронного обмена сообщениями между модулями Программы в брокере сообщений Apache ActiveMQ Artemis должны быть созданы следующие connection factory:

- jms/NSD-CF-CONSUMER;
- jms/NSD-CF-PRODUCER;
- jms/NSD-CF;
- jms/NSD-EVENT\LOGGING\CF;
- jms/NSD-BUSINESS\LOGGING\CF;

и следующие очереди:

- jms/NSD-PORTAL-PROCESSING-DISPATCHER-QUEUE;
- jms/NSD-PUBLIC-PROCESSING-DISPATCHER-QUEUE;
- jms/NSD-PROCESSING-TRANSPORT-QUEUE;
- jms/NSD-REPLICATE\PORTAL\TO_PUBLIC;
- jms/NSD-REPLY\PUBLIC\TO_PORTAL;
- jms/NSD-REPLICATE\PUBLIC\TO_PORTAL;
- jms/NSD-REPLY\PORTAL\TO_PUBLIC;
- jms/NSD-REPLICATE\PORTAL\TO_ARCHIVE;
- jms/NSD-REPLY\ARCHIVE\TO_PORTAL;
- jms/NSD-EVENT\LOGGING\QUEUE;
- jms/NSD-BUSINESS\LOGGING\QUEUE;
- jms/NSD-PORTAL-PROCESSING-DISPATCHER-PRIORITY-QUEUE;

2.4.3 Настройки БД Postgres

Для функционирования ПО E-voting в СУБД PostgreSQL необходимо создать следующие базы данных:

portal_db – БД подсистемы «Обработка персональных данных»

public_db – БД подсистемы «Обработка обезличенных данных»

archive – БД подсистемы «архив»

Должны быть созданы учетные записи НУЗ для БД в контуре:

ta\evote\

ta\archive\evt_

2.4.4 Настройки проксирования nginx

В таблице 5 приведены правила проксирования и маршрутизации HTTP-запросов в Nginx для каждого модуля Программы.

Таблица 5 – Настройки проксирования в nginx

Модуль	Настройки проксирования
*	https://online.e-vote.ru/ □
*	https://online.e-vote.ru/resources □ \${project_path}/common/web-resources/target-grunt
idp	https://online.e-vote.ru/idp/resources □ \${project_path}/common/web-resources/target-grunt
idp	https://online.e-vote.ru/idp □
idp	https://online.e-vote.ru/idp/static □ \${project_path}/idp/app/signin/target-npm
idp	https://online.e-vote.ru/idp/sso/static □ \${project_path}/idp/app/signin/target-npm
idp	https://online.e-vote.ru/idp/signup/static □ \${project_path}/idp/app/signup/target-npm
idp	https://online.e-vote.ru/idp/profile/static □ \${project_path}/idp/app/profile/target-npm
portal	https://online.e-vote.ru/portal □

public	https://online.e-vote.ru/public/files ☐ deny all
public	https://online.e-vote.ru/public/static ☐ \${project_path}/public/public-app/target-npm
public	https://online.e-vote.ru/public/rs ☐ https://host:port/public
public	https://online.e-vote.ru/files ☐ https://host:port/public
portal	https://online.e-vote.ru/interop ☐
portal	https://online.e-vote.ru/emulation ☐ http://host:port/portal_emulation
archive	https://online.e-vote.ru/archive ☐ https://host:port/archive
*	https://online.e-vote.ru/redirect ☐ 404
*	https://online.e-vote.ru/rules ☐ \${project_path}/common/web-resources/target-grunt/html/
*	https://online.e-vote.ru/help ☐ \${project_path}/common/web-resources/target-grunt/html/
idp	☐ https://host:port/idp_core
portal	https://int.online.e-vote.ru/portal ☐ https://host:port/portal_core

public	https://int.online.e-vote.ru/public □ https://host:port/public_core
report	https://int.online.e-vote.ru/report\portal □ https://host:port/report\portal
report	https://int.online.e-vote.ru/report\public □ public>
archive	https://int.online.e-vote.ru/archive □ https://host:port/archive_core

2.4.5 Настройки redis

In-memory хранилище Redis должно быть развернуто в среде виртуализации Kubernetes.

2.5 Порядок установки

2.5.1 Установка базового ПО

2.5.1.1 Разворачивание ПО на серверах с ArtemisMQ

Необходимые файлы и дистрибутивы для разворачивания:

apache-artemis-2.13.0.tar.gz - <https://archive.apache.org/dist/activemq/activemq-artemis/2.13.0/>

artemismq2.tar.gz

jre-8u201-linux-x64.tar.gz - дистрибутив Java

Создаем необходимые директории:

```
mkdir -p /u01/app2 /u01/java /u01/log/artemis/archive
```

/u01/app2 - каталог в который будет установлен ArtemisMQ

/u01/java - Java

/u01/log - каталог в который будут записываться и храниться логи

Добавляем системного пользователя, под которым будет работать сервис Артемиса без права входа:

```
sudo useradd -b /u01 -s /usr/sbin/nologin app2
```

Распаковываем подготовленные архивы в их целевые директории:

```
tar xvzf jre-8u201-linux-x64.tar.gz -C /u01/java/
```

```
tar xvzf artemismq2.tar.gz -C /u01/app2/
```

```
tar xvzf apache-artemis-2.13.0.tar.gz -C /u01/app2/
```

Создаем софт линки:

```
ln -s /u01/java/jre1.8.0_201/ /u01/java/latest
```

```
ln -s /u01/app2/apache-artemis-2.13.0/ /u01/app2/apache-artemis
```

```
ln -s /u01/java/ /usr/java
```

Создаем скрипт java.sh для установки переменных окружения для пользователей:
`vi /etc/profile.d/java.sh`

и записываем туда блок кода:

```
#!/bin/bash
export JAVA_HOME=/u01/java/latest
export JRE_HOME=$JAVA_HOME
export PATH=$PATH:$JAVA_HOME/bin
```

Делаем файл исполняемым, подгружаем его для текущего пользователя и проверяем версию Java:

```
chmod +x /etc/profile.d/java.sh
source /etc/profile.d/java.sh
java -version
```

Создаем файл сервиса SystemD для артемиса:
`sudo vi /etc/systemd/system/artemis.service`

Записываем туда содержимое:

```
[Unit]
Description=Apache ActiveMQ Artemis
After=network.target

[Service]
User=app2
Group=app2
Type=forking
Environment=JAVA_HOME=/u01/java/latest
ExecStart=/u01/app2/artemismq2/bin/artemis-service start
ExecStop=/u01/app2/artemismq2/bin/artemis-service stop
Restart=on-failure
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Далее необходимо поправить пути к директориям и имена серверов в конфигурационных файлах в директории /u01/app2/artemismq2/etc

```
broker.xml
artemis.pro file
bootstrap.xml
logging.properties
а также в директории /u01/app2/artemismq2/bin
artemis
```

Назначаем владельцем директорий служебного пользователя артемис:
`chown -R app2: /u01/app2/ /u01/log/`

Обновляем конфигурацию SystemD и запускаем сервис артемис:
`systemctl daemon-reload`

```
systemctl enable --now artemis.service
```

2.5.1.2 Разворачивание ПО на серверах с Tomcat (WRK, CH)

Необходимые файлы и дистрибутивы для разворачивания:

apache-tomcat-10.0.27.tar.gz - <https://archive.apache.org/dist/tomcat/tomcat-10/v10.0.27/>

NSD_HOME.tar.gz - архив директории с конфигами

openjdk-11.0.2_linux-x64_bin.tar.gz - дистрибутив Java

Создаем необходимые директории:

```
mkdir -p /u01/app1 /u01/java /u01/log/app1/tomcat/archive
```

/u01/app1 - каталог в который будет установлен Tomcat

/u01/java - Java

/u01/log - каталог в который будут записываться и храниться логи Tomcat и приложений

Добавляем системного пользователя, под которым будет работать сервис Томката без права входа:

```
sudo useradd -b /u01 -s /usr/sbin/nologin app1
```

Распаковываем подготовленные архивы в их целевые директории:

```
tar -xvf openjdk-11.0.2_linux-x64_bin.tar.gz -C /u01/java/
```

```
tar -xvf NSD_HOME.tar.gz -C /u01/app1/
```

```
tar -xvf apache-tomcat-10.0.27.tar.gz -C /u01/app1/
```

```
tar -xvf agents.tar.gz -C /u01/
```

Создаем софт линки:

```
ln -s /u01/java/jdk-11.0.2 /u01/java/latest
```

```
ln -s /u01/app1/apache-tomcat-10.0.27 /u01/app1/tomcat
```

```
ln -s /u01/java/ /usr/java
```

Создаем скрипт java.sh установки переменных окружения для пользователей:

```
vi /etc/profile.d/java.sh
```

и записываем туда блок кода:

```
#!/bin/bash
```

```
export JAVA_HOME=/u01/java/latest
```

```
export JRE_HOME=$JAVA_HOME
```

```
export PATH=$PATH:$JAVA_HOME/bin
```

Делаем файл исполняемым, подгружаем его для текущего пользователя и проверяем версию Java:

```
chmod +x /etc/profile.d/java.sh
```

```
source /etc/profile.d/java.sh
```

```
java -version
```

Создаем файл сервиса SystemD для Tomcat:

```
sudo vi /etc/systemd/system/tomcat.service
```

Записываем туда содержимое:

```
[Unit]
```

```
Description=Apache Tomcat Web Application Container
```

```
After=syslog.target network.target
```

```
[Service]
```

```
Type=forking
```

```
User=app1
```

```
Environment=JAVA_HOME=/u01/java/latest
```

```
Environment=CATALINA_PID=/u01/app1/work/tomcat.pid
```

```
Environment=CATALINA_HOME=/u01/app1/tomcat
```

```
Environment=CATALINA_BASE=/u01/app1/tomcat
```

```
ExecStart=/u01/app1/tomcat/bin/startup.sh
```

```
ExecStop=/u01/app1/tomcat/bin/shutdown.sh
```

```
Restart=on-failure
```

```
RestartSec=10
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
Alias=app1.service
```

Проверить пути к директориям и имена хостов в конфигурационных файлах
/u01/app1/NSD_HOME, а также в директории /u01/app1/apache-tomcat-10.0.27/

```
bin/setenv.sh
```

```
conf/context.xml (brokerURL)
```

```
conf/logback.xml
```

```
conf/logback-access.xml
```

```
conf/logging.properties
```

Назначаем владельцем директорий служебного пользователя Tomcat:

```
chown -R app1: /u01/app1/ /u01/log/
```

Проверка лицензии КриптоПро (для чистой установки будет действовать триальная лицензия на 3 месяца):

```
java -cp /u01/app1/tomcat/lib/* ru.CryptoPro.JCP.tools.License
```

Установка лицензии КриптоПро (при наличии номера лицензии) под root и под пользователем приложения:

```
java -cp /u01/app1/tomcat/lib/* ru.CryptoPro.JCP.tools.License -serial "XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX" -company "NRD" -store
```

```
sudo -u app1 /u01/java/latest/bin/java -cp /u01/app1/tomcat/lib/*  
ru.CryptoPro.JCP.tools.License -serial "XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX" -  
company "NRD" -store
```

Обновляем конфигурацию SystemD и запускаем сервис Tomcat:

```
systemctl daemon-reload  
systemctl enable --now tomcat.service
```

2.5.1.3 Разворачивание ПО на серверах с Tomcat (SRV)

Необходимые файлы и дистрибутивы для разворачивания - те же, что и в предыдущем блоке плюс:

www.tar.gz - заархивированная директория /var/www фронта
cacerts - хранилище доверенных сертификатов Java
evoting-nginx-variables - конфиг Nginx
ilb.conf - конфиг Nginx

Делается всё то же, что и в предыдущем разделе для серверов WRK, CH + дополнительные действия, описанные ниже:

Создание директорий для контейнеров ключей (из скрипта инициализации КриптоПро JCP):

```
sudo mkdir -p /var/opt/cprocsp/keys /var/opt/cprocsp/tmp  
sudo chmod a+rwX /var/opt/cprocsp/keys /var/opt/cprocsp/tmp
```

Скопировать cacerts в директорию /u01/java/latest/lib/security сервера с перезаписью:

```
sudo cp cacerts /u01/java/latest/lib/security
```

Установить Nginx для работы внутреннего балансировщика:

```
sudo dnf install nginx
```

Посмотреть выделенный под кэш раздел если он есть (df -h | grep cache), создать там директорию для кеша Nginx:

```
mkdir /var/cache/nginx  
chown -R nginx: /var/cache/nginx
```

Скопировать файлы evoting-nginx-variables и ilb.conf в директорию /etc/nginx/conf.d:

```
sudo cp evoting-nginx-variables ilb.conf /etc/nginx/conf.d
```

Проверить пути к директориям и имена хостов в файлах nginx.conf, evoting-nginx-variables и ilb.conf, отредактировать на новые

Распаковать корневую директорию Nginx со старого сервера:

```
tar -xvf www.tar.gz -C /var
```

Проверить синтаксис конфигов и при отсутствии ошибок запустить Nginx:

```
nginx -t  
systemctl enable --now nginx
```

2.5.1.4 Разворачивание ПО на серверах с Tomcat (WEB)

Необходимые файлы и дистрибутивы для разворачивания похожи на те, что используются на других серверах Tomcat. Отличительная особенность WEB - здесь работают два томката.

- app1-apache-tomcat-10.0.27.tar.gz - tomcat
- app3-apache-tomcat-10.0.27.tar.gz - tomcat
- app1_NSD_HOME.tar.gz - заархивированная директория с конфигами
- app3_NSD_HOME.tar.gz - заархивированная директория с конфигами
- openjdk-11.0.2_linux-x64_bin.tar.gz - дистрибутив Java
- 9O6XWLDD.000\ - директория с контейнерами ключей ЕСИА
- Cacerts - хранилище доверенных сертификатов Java
- www.tar.gz - архив директории /var/www фронта

Создаем необходимые директории:

```
mkdir -p /u01/app{1,3} /u01/java /u01/log/app{1,3}/tomcat/archive
```

Добавляем системных пользователей, под которыми будет работать сервисы Томката:

```
sudo useradd -b /u01 -s /usr/sbin/nologin app1  
sudo useradd -b /u01 -s /usr/sbin/nologin app3
```

Распаковываем подготовленные архивы в их целевые директории:

```
tar -xvf openjdk-11.0.2_linux-x64_bin.tar.gz -C /u01/java/  
tar -xvf app1_NSD_HOME.tar.gz -C /u01/app1/  
tar -xvf app3_NSD_HOME.tar.gz -C /u01/app3/  
tar -xvf app1-apache-tomcat-10.0.27.tar.gz -C /u01/app1/  
tar -xvf app3-apache-tomcat-10.0.27.tar.gz -C /u01/app3/  
tar -xvf agents.tar.gz -C /u01/
```

Делаем софт линки:

```
ln -s /u01/java/jdk-11.0.2 /u01/java/latest  
ln -s /u01/app1/apache-tomcat-10.0.27 /u01/app1/tomcat  
ln -s /u01/app3/apache-tomcat-10.0.27 /u01/app3/tomcat  
ln -s /u01/java/ /usr/java
```

Создаем скрипт java.sh установки переменных окружения для пользователей аналогично предыдущим:

```
vi /etc/profile.d/java.sh
```

записываем туда блок кода:

```
#!/bin/bash  
export JAVA_HOME=/u01/java/latest
```

```
export JRE_HOME=$JAVA_HOME
export PATH=$PATH:$JAVA_HOME/bin
```

Делаем файл исполняемым, подгружаем его для текущего пользователя и проверяем версию Java:

```
chmod +x /etc/profile.d/java.sh
source /etc/profile.d/java.sh
java -version
```

Проверить пути к директориям и имена хостов в конфигурационных файлах /u01/app1/NSD_HOME, /u01/app3/NSD_HOME, а также в директориях /u01/app1/apache-tomcat-10.0.27/, /u01/app3/apache-tomcat-10.0.27/

```
bin/setenv.sh
conf/context.xml (brokerURL)
conf/logback.xml
conf/logback-access.xml
conf/logging.properties
```

Назначаем владельцем директорий служебных пользователей Tomcat:

```
sudo chown -R app1: /u01/app1/ /u01/log/app1/
sudo chown -R app3: /u01/app3/ /u01/log/app3/
```

Установка лицензии КриптоПро (при наличии номера лицензии) под root и под пользователями приложений:

```
java -cp :/u01/app1/tomcat/lib/* ru.CryptoPro.JCP.tools.License -serial
"XXXXXX-XXXXX-XXXXX-XXXXX-XXXXX" -company "NRD" -store
sudo -u app1 /u01/java/latest/bin/java -cp :/u01/app1/tomcat/lib/*
ru.CryptoPro.JCP.tools.License -serial "XXXXXX-XXXXX-XXXXX-XXXXX-XXXXX"
-company "NRD" -store
sudo -u app3 /u01/java/latest/bin/java -cp :/u01/app3/tomcat/lib/*
ru.CryptoPro.JCP.tools.License -serial "XXXXXX-XXXXX-XXXXX-XXXXX-XXXXX" -company "NRD"
```

Создание директорий для контейнеров ключей:

```
sudo mkdir -p /var/opt/cprocsp/keys/app3 /var/opt/cprocsp/tmp
sudo chmod a+rwX /var/opt/cprocsp/keys /var/opt/cprocsp/tmp
```

Скопировать туда папку с контейнером ЕСИА 906XLDD.000\ и назначить ей владельца приложения App3:

```
sudo cp -r 906XLDD.000/ /var/opt/cprocsp/keys/app3
sudo chown -R app3: /var/opt/cprocsp/keys/app3
```

Скопировать cacerts в директорию /u01/java/latest/lib/security сервера с перезаписью:

```
sudo cp cacerts $JAVA_HOME/lib/security/
```

Создаем файл сервиса SystemD для Tomcat:

```
sudo vi /etc/systemd/system/tomcat.service
```

Записываем туда содержимое:

[Unit]

Description=Apache Tomcat Web Application Container

After=syslog.target network.target

[Service]

Type=forking

User=app1

Environment=JAVA_HOME=/u01/java/latest

Environment=CATALINA_PID=/u01/app1/work/tomcat.pid

Environment=CATALINA_HOME=/u01/app1/tomcat

Environment=CATALINA_BASE=/u01/app1/tomcat

ExecStart=/u01/app1/tomcat/bin/startup.sh

ExecStop=/u01/app1/tomcat/bin/shutdown.sh

Restart=on-failure

RestartSec=10

[Install]

WantedBy=multi-user.target

Alias=app1.service

sudo vi /etc/systemd/system/tomcat3.service

Записываем туда содержимое:

[Unit]

Description=Apache Tomcat Web Application Container

After=syslog.target network.target

[Service]

Type=forking

User=app3

Environment=JAVA_HOME=/u01/java/latest

Environment=CATALINA_PID=/u01/app3/work/tomcat.pid

Environment=CATALINA_HOME=/u01/app3/tomcat

Environment=CATALINA_BASE=/u01/app3/tomcat

ExecStart=/u01/app3/tomcat/bin/startup.sh

ExecStop=/u01/app3/tomcat/bin/shutdown.sh

Restart=on-failure

RestartSec=10

[Install]

WantedBy=multi-user.target

Alias=app3.service

делаем reload для SystemD и запускаем сервисы:

sudo systemctl daemon-reload

sudo systemctl enable --now tomcat.service

sudo systemctl enable --now tomcat3.service

Устанавливаем Nginx для балансировки входящего трафика и отображения фронта (если есть отдельный сервер NLB, эту часть нужно выполнять там)

```
sudo dnf install nginx
```

Посмотреть выделенный под кэш раздел (df -h |grep cache), создать там директорию для кеша Nginx:

```
sudo mkdir /var/cache/nginx  
sudo chown -R nginx: /var/cache/nginx
```

В nginx.conf добавляем параметр в раздел General options (в RedOS 8 такая опция есть по умолчанию):

```
types_hash_max_size 4096;
```

Скопировать файлы evoting-nginx-variables и ilb.conf в директорию /etc/nginx/conf.d:

```
sudo cp evoting-nginx-variables ilb.conf /etc/nginx/conf.d
```

Также скопировать директорию с сертификатами /etc/nginx/ssl в аналогичную.

Проверить пути к директориям и имена хостов в файлах nginx.conf, evoting-nginx-variables и ilb.conf.

Распаковать корневую директорию Nginx:

```
tar -xvf www.tar.gz -C /var
```

Проверить синтаксис конфигов и при отсутствии ошибок запустить сервис Nginx:

```
nginx -t  
systemctl enable --now nginx
```

2.5.1.5 Установка PostgreSQL

Установка PostgreSQL с помощью менеджера пакетов dnf
dnf install postgresql15-server postgresql15

Инициализация базы данных
/usr/pgsql-15/bin/postgresql-15-setup initdb

Добавляем службу в автозагрузку и запускаем её
systemctl enable postgresql-15 --now

Проверяем статус службы
systemctl status postgresql-15

Задаем пароль для административного пользователя postgres
ALTER USER postgres WITH ENCRYPTED PASSWORD 'yourpassword';

2.5.1.6 Установка Redis

Установка

```
dnf install redis
```

Добавляем службу в автозагрузку и запускаем её

```
systemctl enable --now redis
```

Проверяем статус службы

```
systemctl status redis
```

Проверяем связь

```
redis-cli ping
```

Разрешить подключение к Redis не только с локального хоста

```
nano /etc/redis/redis.conf
```

добавляем необходимые IP-адреса

После сохранения изменений перезапускаем службу

```
systemctl restart redis
```

2.5.2 Развертывание BackEnd

Сборка серверной части выполняется автоматически в системе непрерывной интеграции GitLab. Результат сборки — war-архивы с кодом приложения — выкладывается в репозиторий артефактов Nexus.

Развертывание на контур выполняется с помощью Ansible-скриптов, запускаемых из системы управления автоматизацией AWX.

Основные шаги установки:

1. Ansible-скрипты загружают war-архивы из репозитория Nexus на серверы контура в каталог автодеплойа Apache Tomcat.
2. Выполняется перезапуск службы Apache Tomcat.
3. При старте Apache Tomcat автоматически выполняется развертывание war-архивов.
4. Обновление схемы базы данных выполняется автоматически с помощью Liquibase-скриптов в момент старта сервера приложений Apache Tomcat.

Для корректного развертывания серверной части необходимо предварительное наличие и настройка следующих компонентов: Apache Tomcat, PostgreSQL, Apache ActiveMQ Artemis, Redis, а также создание баз данных и учетных записей в соответствии с подразделом 2.4.3.

2.5.3 Развертывание FrontEnd

Клиентская часть (фронтенд) реализована в виде Angular-приложения. Для развертывания необходим HTTP-сервер для отдачи статического контента (код приложения, разметка, стили, изображения и т.д.) и reverse проху-сервер для переадресации запросов к API бэкенда. В текущей конфигурации обе задачи выполняет Nginx.

Таблица настроек проксирования приведена в разделе «Настройки проксирования» (2.4.4).

Сборка клиентской части выполняется автоматически в системе непрерывной интеграции GitLab. Результат сборки — zip-архив с кодом приложения — выкладывается в репозиторий артефактов Nexus.

Развертывание на контур выполняется с помощью Ansible-скриптов, запускаемых из системы управления автоматизацией AWX.

Основные шаги установки:

1. Ansible-скрипты загружают zip-архив из репозитория Nexus на сервер Nginx.
2. Выполняется перезапуск службы Nginx.

Для корректного развертывания клиентской части необходимо предварительное наличие и настройка веб-сервера Nginx, а также настройка правил проксирования в соответствии с подразделом 2.4.4.

2.5.4 Проверка работоспособности после установки

По журналам логов Tomcat всех контуров проверяется отсутствие ошибок при запуске системы.

Выполняется вход в UI Tomcat.

Выполняется авторизация в административный модуль, в UI проверяется работа всех возможностей модуля.

Выполняется авторизация под тестовым пользователем с помощью Сбер ID и через госуслуги, в UI проверяется корректность работы.

С помощью дашборда мониторинга проверяется успешный запуск и доступность всех компонентов системы.

2.5.5 Удаление/откат

В случае наличия проблем после установки новой версии есть возможность откатиться на предыдущую успешную версию. Для этого в системе управления автоматизацией AWX указывается номер нужной предыдущей версии и выполняется запуск установки, автоматически выполняются шаги указанные в п.п. 2.5.2, 2.5.3

3. ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

Термин/сокращение	Определение/обозначение
Система, E-voting	Информационная система Электронного голосования E-voting
ЦОД	Центр обработки данных
СУБД	Система управления базами данных
ООД	Обработка обезличенных данных
ОПД	Обработки персональных данных
ЦБ	Ценные бумаги
Ценные бумаги	Ценные бумаги
E-proxy voting	Информационная система, предоставляющая владельцам ЦБ возможность голосования в электронном виде на ОС с использованием услуг депозитария
ОЗУ	Оперативная память
НЖМД	Накопитель жесткий диск
ФГИС ЕСИА	Федеральная государственная информационная система Единого учета и аутентификации
СПЭП ИС ГУЦ	Система электронного документооборота и управления корпоративными процессами

Термин/сокращение	Определение/обозначение
ИАС РОП	Информационная автоматизированная система регулирования обращения на рынке ценных бумаг
СМЭВ	Система межведомственного электронного взаимодействия, используемая для обмена данными между различными государственными органами
Антифрод НРД	Система антифрод, используемая для предотвращения мошенничества и обеспечения безопасности при проведении голосования
ИС Регистраторов	Информационная система, используемая для управления списками владельцев ценных бумаг и их правами