Аннотация

Настоящее руководство содержит информацию, необходимую для установки и настройки программного комплекса выявления и блокирования несанкционированного доступа к информационным системам на основе анализа конфликта полномочий с использованием технологий искусственного интеллекта (ПК SoD).

Руководство предназначено для сотрудников компании-заказчика (администраторов), обладающих опытом настройки и управления локальными сетями и оборудованием, установки и настройки программного обеспечения инфраструктуры для автоматизации развёртывания, масштабирования и управления контейнеризированными приложениями.

Используемые в документе термины и определения и сокращения описаны в соответствующих разделах.

Оглавление

1. Требования к установке ПО

Специфика применения программного комплекса подразумевает его использование в составе информационной системы, реализуемой в рамках конкретного проекта.

Процедуры установки ПО на серверы и рабочие места пользователей, проверки работоспособности ПО, настройки серверной и клиентской частей ПО и интеграций объединены и обеспечиваются последовательностью действий, выполняемых по алгоритмам, описанным в соответствующих разделах настоящего документа.

Для выполнения работ по установке и настройке ПО системы необходима настроенная рабочая станция (персональный компьютер) с возможностью подключения к локальной сети, объединяющей предварительно настроенные компоненты инфраструктуры заказчика (серверы с установленными ОС).

1.1. Инсталляционный комплект

Инсталляционный комплект представляет собой набор файлов, определяемый условиями реализации ИС в рамках конкретного проекта. Состав набора определяется условиями технического задания проекта (включая требования к ИС и к инфраструктуре заказчика).

Инсталляционный комплект используется для установки и первичной настройки ИС сотрудниками компании-разработчика.

1.2. Требования к техническому обеспечению рабочих станций

Рабочие станции (персональные компьютеры пользователей и администраторов) должны обладать следующими техническими характеристиками:

- CPU — 2 ядра;
- объём памяти RAM — 4 Гб;
- объём хранения данных — 32 Гб;
- монитор SVGA;
- клавиатура 101/102 клавиши;
- манипулятор типа «мышь»;
- устройство чтения компакт-дисков (DVD-ROM);
- сетевой адаптер 100 Мбит/с.

Программное обеспечение рабочих станций должно работать под управлением ОС Astra Linux 1.7 Орёл (минимальная версия) и позволять использование ПО Microsoft Office 2003 (минимальная версия).

На рабочих станциях должны запускаться актуальные версии одного из следующих веб-браузеров: Mozilla FireFox, Яндекс.Браузер, Google Chrome.

1.3. Требования к техническому обеспечению серверов

Сервер платформы контейнеризации (master):

- количество серверов не менее 3 шт.;
- CPU — 4 ядра;
- объём памяти RAM — 8 Гб;
- объём хранения данных — 200 Гб.

Сервер инфраструктурных сервисов (worker):

- количество серверов не менее 2 шт.;
- CPU — 4 ядра (рекомендовано 16 ядер);
- объём памяти RAM — 8 Гб (рекомендовано 32 Гб);
- объём хранения данных — 200 Гб.

Файловый сервер:

- CPU — 4 ядра;
- объём памяти RAM — 16 Гб
- объём хранения данных — 5 Тб.

Сервер облачного хранилища:

- CPU — 4 ядра;
- объём памяти RAM — 16 Гб;
- объём хранения данных — 532 Гб.

Сервер БД:

- CPU — 32 ядра;
- объём памяти RAM — 96 Гб;
- объём хранения данных — 1 Тб.

Платформа управления репозиториями

- CPU — 4 ядра;
- объём памяти RAM — 8 Гб;
- объём хранения данных — 200 Гб.

1.4. Требования к квалификации администраторов

Профессиональная квалификация администраторов, обеспечивающих эксплуатацию ПК SoD, подразумевает понимание принципов работы по следующим направлениям:

- понимание принципов ведения НСИ;
- достаточные знания в предметной области работы компании-заказчика;

- владение процедурным языком OCL;
- владение языком регулярных выражений;
- знание нотации BPMN;
- владение организацией интеграционного взаимодействия ИС с использованием API.

Квалификация системных администраторов, обеспечивающих эксплуатацию ПК SoD, должна также позволять:

- использовать стандартные возможности применяемых типовых средств вычислительной техники, операционных систем, СУБД и другого системного программного обеспечения;
- работать с архиваторами, дисковыми утилитами, антивирусными программами и программами резервного копирования;
- определять источник сбоя функционирования и отказа ИС;
- восстанавливать работоспособность ИС после сбоя или отказа;
- проводить регламентные работы и обслуживание ИС;
- обеспечивать требуемые условия эксплуатации ИС.

Пользователи-операторы ПК SoD должны обладать знаниями и навыками работы со следующим ПО:

- операционная система Microsoft Windows 7 и выше;
- операционные системы семейства Linux;
- Microsoft Office 2003 и выше.

Описание разделения администраторов по ролям и описание выполняемых ими функций содержатся в следующем разделе.

2. Установка и настройка ПК SoD

Действия по инсталляции ПК SoD не выходят за рамки требований, предъявляемых к пользователям со следующими ролями:

1. Администратор инфраструктурных сервисов.
2. Администратор серверов контейнеризации.

Администратор инфраструктурных сервисов

Администратор инфраструктурных сервисов реализует следующие функции:

- инсталляция и настройка сервера приложений и инфраструктурных сервисов;
- настройка системных компонентов сервера приложений;
- первичные настройки системных компонентов;
- обновление компонентов программного обеспечения;
- контроль работоспособности серверов и сервисов, восстановление работы после технических отказов;
- контроль работоспособности компонентов;
- контроль отказоустойчивости;
- контроль функционирования серверов и сервисов, устранение неполадок.

Администратор серверов контейнеризации

Администратор серверов контейнеризации реализует следующие функции:

- инсталляция и настройка серверов контейнеризации;
- контроль работоспособности контейнеризации, восстановление работы после технических отказов. Восстановление работоспособности выполняется автоматически. В случае невозможности выполнения восстановления работоспособности средствами системы администратору следует обратиться к разработчикам;
- контроль функционирования контейнеризации, устранение неполадок. Устранение неполадок выполняется либо администратором, либо сотрудниками компании-разработчика после соответствующего обращения.

3. Объекты «Планета»

Для реализации, заложенной в ПК SoD функциональности, необходимо использование следующих программных продуктов «Планета»:

- «Планета. НСИ» — реализует функциональность управления справочниками и реестрами;
- «Планета. Доступ» — реализует функциональность управления авторизации, аутентификации и управления ролевой моделью;
- «Планета. Аналитика» — реализует функциональность конфигурирования, формирования и распространения отчетности и информационных панелей;
- «Планета. Процессы» — реализует функциональность конфигурирования и последующего выполнения процессов внесения изменений в основные данные;
- «Планета. Интеграция» — реализует функциональность поддержки процессов загрузки, консолидации и обработки больших объёмов данных в наборы данных для последующего анализа (опционально, в зависимости от конкретного проекта).

3.1. Установка «Планета. НСИ»

Состав и последовательность действий по установке и настройке «Планета. НСИ» зависят от особенностей информационной системы, реализуемой в рамках конкретного проекта (например, от выбора в качестве средства управления контейнерами Docker или Docker Compose).

Необходимые объекты «Планета» и стороннее ПО:

- программный продукт «Планета. Доступ»;
- инструмент оркестрации K8s (+Deckhouse);
- СУБД PostgreSQL;
- менеджер репозиториев Nexus (опционально, в зависимости от конкретного проекта).

Выполняемые действия:

- подготовка микросервисов (share);
- проверка наличия секрета доступа к репозиторию;
- изменение значений в /r-28-1-nsi-backend-config-maps.TO_FILL_UP.yaml;
- изменение в файле *r-30-ingress-rules.yaml* доменов вида *.dex-m.planeta.w18* на необходимые;
- применение всех остальных манифестов;
- подготовка БД (создание БД, пользователя, выдача прав пользователю на БД);

- изменение конфигурационных файлов;
- настройка сервиса верификации данных в справочниках НСИ.

Пример порядка развёртывания компонента «Планета. НСИ»

Внимание!

Следует учитывать зависимость содержания выполняемых скриптов от конкретной версии «Планета. НСИ».

Способ указания параметров для FQDN-сервиса зависит от конкретного проекта.

1. Создать директорию проекта с помощью команды:

   ```
   mkdir -p Путь / директория проекта
   ```

2. Перенести в созданную директорию архивы с docker-образами и с файлами конфигурации для запуска сервисов:

   ```
   cp -R / директория с архивами / директория проекта
   ```

3. Разархивировать приложенные архивы, перейти в директорию images и загрузить docker-образы с помощью команды:

   ```
   sh restore.sh
   ```

4. Проверить загруженные docker-образы с помощью команды:

   ```
   docker image ls
   ```

5. Перейти в директорию traefik и запустить docker-контейнер *traefik gateway* с помощью команды:

   ```
   docker compose up -d
   ```

6. Убедиться, что запущенный контейнер имеет статус UP с помощью команды:

   ```
   docker-compose ps
   ```

7. Перейти в директорию nsi и запустить контейнеры «Планета. НСИ» с помощью команды:

   ```
   docker up -d
   ```

   Внимание!
   Инициализация базы данных и запуск миграций может занимать длительное время, которое определяется ресурсами сервера.
   Строго не рекомендуется прерывать запуск контейнеров!

8. Убедиться, что запущенные контейнеры «Планета. НСИ» имеют статус **UP** с помощью команды:

   ```
   docker-compose ps
   ```

9. Список компонентов с требуемым статусом должен выглядеть следующим образом:

   ```
   nsi_access-oauth_1
   nsi_access-pg_1
   nsi_analytics-pg_1
   ```

```
nsi_config_1
nsi_content_1
nsi_discovery_1
nsi_gateway_1
nsi_nsi-backend_1
nsi_nsi-frontend_1
nsi_nsi-pg_1
nsi_redis_1
```

3.2. Установка «Планета. Доступ»

Необходимые компоненты стороннего ПО:

- инструмент оркестрации K8s (Deckhouse);
- инструмент кеширования Redis;
- СУБД PostgreSQL.

Выполняемые действия:

- запрос и изменение пароля платформы контейнеризации для учётной записи;
- добавление пользователя в БД;
- настройка Redis для «Планета. Доступ»;
- подготовка общих сервисов для компонентов «Планета»;
- применение всех остальных манифестов.

Пример порядка развёртывания компонента «Планета. Доступ» для Linux (docker)

Подготовка серверного окружения

1. Войти на удаленную машину по ssh (либо открыть локальный терминал).
2. Убедиться, что установлен необходимый софт:

Команда	Назначение	Ожидаемый вывод
`git --version`	проверка наличия git клиента	`git version 2.*.*`
`docker info`	проверка наличия docker	`Server Version: 20.*.*`
`docker-compose -v`	проверка наличия docker-compose	`docker-compose version 1.2*.*, build *`

Сбор данных о системе

Для начала настройки потребуется:

1. Узнать основной IP-адрес первичного сетевого интерфейса (например, это можно сделать командами `ip a` или `ifconfig`.
2. Создать локальное доменное имя в DNS (или в файле /etc/hosts). Например, это будет *planeta.ibs*.
3. Нужно убедиться, что все поддомены локального домена также разрешаются.
4. В случае, если используется DNS, можно создать запись типа CNAME для **.planeta.ibs* со значением **planeta.ibs**.
5. Если используется файл /etc/hosts, требуется явным образом перечислить все субдомены: *admin.planeta.ibs*, *registry.planeta.ibs*, *swagger.planeta.ibs*, *traefik.planeta.ibs*.

Подготовка конфигурации «Планета. Доступ»

1. Создать директорию для размещения файлов дистрибутива проекта «Планета. Доступ».
2. Отредактировать в файле docker-compose.yaml следующие переменные:

Переменная	Значение	Пример
LOCAL_DOMAIN	имя локального домена	planeta.ibs
LOCAL_PORT	TCP порт на сервере (должен быть свободен)	80 Изменять значение по умолчанию есть смысл только если порт 80 уже занят другим приложением
LOCAL_IP	основной адрес первичного интерфейса текущей машины (не должен быть 127.0.0.1). Можно указать адрес сети, созданной Docker	172.17.0.1

3. Получить доступ к хранилищу docker registry и авторизоваться. Логин и пароль выдаются администраторами проекта «Планета»:

Команда	Назначение	Ожидаемый вывод
`docker login harbor.nexus.planetanalytics.ru`	вход в удаленный репозиторий Docker	`Login Succeeded`
`./dc pull`	получение образов	`done`
`./dc up -d`	запуск контейнеров	`unhealthy`
`./dc config > config.yml`	получение единого yaml-файла docker-compose). Этот шаг позволяет получить линейным текстом конфигурацию	`config.yml`

Команда	Назначение	Ожидаемый вывод
	маршрутов и переменных ENV, чтобы видеть полную картину связей, настроек и зависимостей	

4. При первом запуске «Планета. Доступ» необходимо добавить пространство, пользователя, клиента и список ролей. Эти действия описаны в документах Руководство администратора «Планета. Доступ» и Руководство пользователя «Планета. Доступ».
5. Открыть в браузере ссылку http://admin.planeta.ibs (если порт по умолчанию изменялся, то нужно явно указать порт).
6. Авторизоваться как пользователь admin с паролем admin (при первом входе будут сообщения об ошибке, которые нужно игнорировать).
7. Создать клиента с именем **content-service** и указать для него:

   метод аутентификации:

   ```
   CLIENT_SECRET_POST
   CLIENT_SECRET_BASIC
   ```

   метод получения токена:

   ```
   CLIENT_CREDENTIALS
   ```

8. Активировать клиента.
9. Перейти в консоль и повторно выполнить команду: `./dc up -d`.
10. Убедиться, что после **п. 6** content service также успешно запустился.
11. Обновить страницу браузера и убедиться, что ошибки пропали.

Пример порядка развёртывания компонента «Планета. Доступ»

Внимание!

В настоящее время доступен вариант скрипта запуска развёртывания одной командой (см. https://bb.ibs.ru/projects/DEVOPS/repos/deckhouse-k8s/browse/planeta-k8s-helm-chart/readme.md?at=planeta-shared-k8s-services).

1. Создать namespace planeta:

```
kubectl create ns planeta
```

2. Запросить Docker-пароль для учетной записи.

Изменить docker-password и docker-email:

```
kubectl create secret -n planeta docker-registry ibs-planeta-registry --docker-server=planeta-release.nexus.x-dev.us --docker-username=release- ibs --docker-password=PUT-HERE-YOUR-PASSWORD --docker-email=PUT-HERE@YOUR-EMAIL
```

3. Добавить пользователя в БД с помощью скриптов для Postgres (пример):

```
CREATE USER "postgres-access-u" WITH PASSWORD 'PUT-HERE-
THE-PASSWORD';
CREATE DATABASE "postgres-access-db";
GRANT ALL ON DATABASE "postgres-access-db" TO "postgres-
access-u";
\c "postgres-access-db"
CREATE EXTENSION IF NOT EXISTS pgcrypto;
CREATE EXTENSION pgcrypto;
```

Выполнить стартовые скрипты в БД (realm ibs изменить на необходимый):

- sql-migration-01-create-realm.sql;
- sql-migration-02-add-user.sql;
- sql-migration-03-add-client.sql.

4. Изменить в r-04-config-maps.yaml параметры Postgres connection.
5. Добавить пароль для пользователя БД:

```
kubectl -n planeta create secret generic postgres --from-
literal=PG_DB_PASS=PUT-PASSWORD-HERE
```

6. Настроить Redis для «Планета. Доступ».

После установки Redis необходимые параметры по умолчанию содержатся в репозитории. В случае необходимости их можно переопределить.

7. Настроить Shared K8s Services.

Изменить соответствующие параметры:

- ELK (Host: elk.planeta.ibs, port: 5001);
- хосты вида *.dex-m.planeta.w18 на необходимые.

Применить все файлы в папке ../planeta-shared-k8s-services/:

```
kubectl -n planeta apply -f ../planeta-shared-k8s-sesrvices/
```

8. Поднять элементы приложения «Планета. Доступ»

Последовательно выполнить манифесты r-0X, r-1X, frontend:

```
kubectl -n planeta apply -f ./r-0X...
kubectl -n planeta apply -f ./r-11-...
kubectl -n planeta apply -f ./frontend-...
```

«Планета. Доступ» доступна по одному из Ingress endpoints:

```
kubectl -n planeta get ingress planeta-access-routes
```

9. Выполнить обновление «Планета. Доступ»

Исходное состояние (файл r-11-oauth-deployment.yaml):

```
image: planeta-release.nexus.x-dev.us/access-docker-
repository/release/release-1.1/auth-service:latest
```

```
kubectl -n: planeta scale deploy/access-oauth --replicas=0
# Замена Image 1.1-1.2
image:        planeta-release.nexus.x-dev.us/access-docker-repository/release/release-1.2/auth-service:latest
#Scale
kubectl -n: planeta scale deploy/access-oauth --replicas=1
# Стабилизация, далее
# Повторить алгоритм для 1.2->1.3
```

4. Стороннее программное обеспечение

Для установки, развёртывания и применения ПК SoD необходимо использовать программное обеспечение сторонних производителей.

Состав стороннего ПО и реализуемая им функциональность зависят от конкретного проекта.

Пример набора стороннего ПО, обеспечивающего работу ПК SoD в составе ИС на основе «Планета. НСИ»:

- средство контейнеризации и оркестрации — Kubernetes (Deckhouse);
- приложение балансировки нагрузки — Kubernetes;
- файловый сервер — распределенная файловая система NFS;
- сервер облачного хранилища — MinIO;
- система управления базами данных — Postgres;
- платформа управления репозиториями — Nexus.

Сведения об установке и настройке стороннего ПО содержатся в документе Руководство по установке и настройке ПК SoD.

4.1. Установка стороннего ПО

Набор стороннего ПО, реализуемая им функциональность и, соответственно, последовательность действий по установке и настройке зависят от особенностей информационной системы, реализуемой в рамках конкретного проекта.

5. Обновление ПО

Обновление операционной системы и стороннего ПО, обеспечивающих функционирование ИС выполняется в соответствии с политиками производителей конкретных ОС и стороннего ПО.

Сведения о поддержке программного продукта содержатся в документе Руководство пользователя ПК SoD.

6. Набор документации ПК SoD

Общие сведения о продукте содержатся в документе Описание программы ПК SoD.

Сведения о поддержке программного продукта содержатся в документе Руководство администратора ПК SoD.

Сведения об использовании продукта содержатся в документе Руководство пользователя ПК SoD.

Термины и определения

Термин	Определение
Администратор	Сотрудник организации, обладающий компетенциями и правами для управления ИС
Платформа «Планета»	Набор программных продуктов, разработанных ГК ИБС, и обеспечивающих решение различных корпоративных задач в области обработки данных и управления данными и процессами компании
Компонент (программный компонент)	Часть программного обеспечения ИС, выполняющая определённые задачи
Программный продукт	Набор программных компонентов
Справочник	Сущность ИС, содержащая определённые данные и позволяющая их обработку

Сокращения

Сокращение	Значение
БД	База данных
ИС	Информационная система
НСИ	Нормативно-справочная информация
ПО	Программное обеспечение
СУБД	Система управления базами данных
ELK	Elasticsearch, Logstash, Kibana — программные решения для решения задач поиска и аналитики при обработке данных