Инструкция по установке (развертыванию экземпляра на сервере) экземпляра ПО

Программное обеспечение ЭВМ «Cicada8 Dependency Firewall»

CIOADA 8

Оглавление

1. Введение

Настоящий документ содержит описание действий по установке и началу работы с Программным обеспечением ЭВМ «Cicada8 Dependency Firewall».

Программное обеспечение ЭВМ «Cicada8 Dependency Firewall» (далее – Система или Dependency Firewall) представляет собой программное обеспечение, включающее в себя совокупность интерфейса и автоматизированных сервисов, направленных на анализ безопасности компонентов и зависимостей программных решений.

2. Переменные окружения и описание репозиториев Dependency Firewall в GitLab

2.1 Переменные окружения для сервиса Firewall

Для работы с Системой необходимо задать переменные окружения сервиса Firewall (все переменные являются обязательными), представленные в таблице 1.

Таблица 1 – Переменные окружения сервиса Firewall

Переменная	Описание	Значение
FIREWALL_CONFIG_EP	Адрес получения конфигурационных файлов	https://dep-fw.local/api/firewalls/get-config/
FIREWALL_REGISTRATION_EP	Адрес регистрации	https://dep-fw.local/api/firewalls/registration/
FIREWALL_TOKEN	Токен для работы с Firewall	43b5698d-befb-4fa8-a4e2-424a0f9c4d0116:09

Все остальные переменные окружения будут автоматически переданы со стороны бэкенда при успешной регистрации.

2.2 Переменные окружения для сервиса Feeder

Сервис может быть настроен с помощью переменных окружения. Список всех настраиваемых параметров представлен в таблице 2.

Таблица 2 – Список настраиваемых параметров

Переменная	Описание	Пример значения	Обязательная переменная
Конфигурация приложения			
FEEDER_SERVER_PORT	Порт для HTTP-сервера	8089	Да
FEEDER_SERVER_MAX_CLIENTS	Максимальное количество одновременно работающих клиентов	100	Да
FEEDER_SERVER_MAX_UPLOAD_FILE_SIZE	Максимальный размер (в мегабайтах) загружаемых файлов	2000	Да
Таймауты HTTP-сервера			
FEEDER_SERVER_PORT	Порт для HTTP-сервера	8089	Да
FEEDER_SERVER_MAX_READ_TIMEOUT	Максимальная продолжительность	20m	Нет

	чтения запроса, включая тело запроса		
FEEDER_SERVER_MAX_WRITE_TIMEOUT	Максимальная продолжительность до завершения записи ответа	20m	Нет
FEEDER_SERVER_MAX_IDLE_TIMEOUT	Максимальное время ожидания следующего запроса при включенной функции keep-alives	30m	Нет
Конфигурация логгирования			
FEEDER_LOG_LEVEL	Уровень логирования	debug	Нет
Хранение данных			
FEEDER_STORAGE_PATH	Путь, где локально хранятся файлы данных	data	Да
Конфигурация Docker Registry			
FEEDER_REGISTRY_URL	URL-адрес Docker Registry, где хранится база данных	registry.cicada8.ru/feeds/db:latest	Да
FEEDER_REGISTRY_USER	Имя пользователя для доступа к Docker Registry	user	Нет
FEEDER_REGISTRY_PASS	Пароль для доступа к Docker Registry	pass	Нет
Конфигурация прокси			
FEEDER_PROXY_URL	URL-адрес Proxy registry (upstream proxy)	http://proxy.ru:3128	Нет
FEEDER_PROXY_AUTH	Заголовок авторизации для прокси в формате base64	bXlVc2lybmFtZTpteVBhc3N3b3JkMTIzNA==	Нет
Конфигурация Redis			
FEEDER_REDIS_ADDRESS	Адрес сервера Redis	localhost:6379	Да
FEEDER_REDIS_PASSWORD	Пароль для сервера Redis	pass	Нет
FEEDER_REDIS_DB	Номер базы данных Redis	1	Нет
FEEDER_REDIS_STREAM	Поток Redis, в который отправляются сигналы обновления	app:feeder	Да

FEEDER_REDIS_TIMEOUT	Таймаут для соединений Redis	5s	Да
Плановый интервал			
FEEDER_SCHEDULER_INTERVAL	Интервал, с которым служба проверяет наличие обновлений в Docker Registry	5m	Да

2.3 Переменные окружения для бэкенда

Переменные окружения для бэкенда представлены в таблице 3.

Таблица 3 – Список настраиваемых параметров

Переменная	Тип	Описание	Значение по умолчанию	Обязательная
ALLOWED_HOSTS	str	Разрешенные хосты		Да
APPSECHUB_INTEGRATION_ENABLED	bool	Интеграция с appsec	False	Нет
AWS_ACCESS_KEY_ID	str	Ключ для доступа в Minio		Да
AWS_DEFAULT_REGION	str	Регион для Minio	us-east-1	Нет
AWS_LOCATION	str	Папка в Minio для статических файлов Django	static	Нет
AWS_PRIVATE_MEDIA_LOCATION	str	Папка для приватных файлов пользователя	private-media	Нет
AWS_PUBLIC_MEDIA_LOCATION	str	Папка для публичных файлов пользователя	media	Нет
AWS_QUERYSTRING_AUTH	bool	Использование query-параметров для аутентификации	True	Нет
AWS_S3_ENDPOINT_URL	str	URL хранилища S3		Да
AWS_S3_URL_PROTOCOL	str	Протокол для S3	http:	Нет
AWS_SECRET_ACCESS_KEY	str	Секретный ключ для доступа в Minio		Да
AWS_STATIC_DOMAIN	str	Домен Minio		Да
AWS_STORAGE_BUCKET_NAME	str	Название бакета в Minio		Да
CORS_ALLOWED_ORIGINS	str	Ресурсы, с которых разрешены запросы		Да

Переменная	Тип	Описание	Значение по умолчанию	Обязательная
DB_HOST	str	Хост базы данных		Да
DB_NAME	str	Название базы данных		Да
DB_PASSWORD	str	Пароль к базе данных		Да
DB_PORT	int	Порт для соединения с БД	5432	Нет
DB_USER	str	Пользователь БД		Да
DEBUG	bool	Режим отладки	False	Нет
DJANGO_ENV	str	Окружение	development	Нет
FIREWALL_AWAIT_CONCURRENCY	int	Количество асинхронных операций	10	Нет
FIREWALL_AWAIT_TIMEOUT	str	Макс. продолжительность асинхронных операций	300s	Нет
FIREWALL_AWAIT_UPDATE	str	Период обновления для асинхронных операций	300ms	Нет
FIREWALL_FEED_ADDR	str	Адрес сервиса для получения фидов	feeder:8089	Нет
FIREWALL_FEED_ENABLED	bool	Включение получения фидов	True	Нет
FIREWALL_FEED_POLL_INTERVAL	str	Интервал между запросами фидов	5m	Нет
FIREWALL_FEED_RETRY_COUNT	int	Количество повторных попыток запроса фидов	15	Нет
FIREWALL_FEED_RETRY_INTERVAL	str	Интервал между повторными попытками	30s	Нет
FIREWALL_FEED_TIMEOUT	str	Таймаут запроса для получения фидов	1m	Нет
FIREWALL_REDIS_ACTIVITIES	str	Стрим activities в Redis	app:packets	Нет
FIREWALL_REDIS_ADDR	str	Адрес Redis	redis:6379	Нет
FIREWALL_REDIS_DB	int	Номер БД Redis	1	Нет
FIREWALL_REDIS_FEED	str	Стрим фидов в Redis	app:feeder	Нет

Переменная	Тип	Описание	Значение по умолчанию	Обязательная
FIREWALL_REDIS_IDENTITIES	str	Стрим identities в Redis	app:identities	Нет
FIREWALL_REDIS_PASS	str	Пароль Redis	some_pass	Нет
FIREWALL_REDIS_POLICIES	str	Стрим политик в Redis	app:policies	Нет
FIREWALL_REDIS_POLICY_RULES	str	Стрим правил политик в Redis	app:firewall-policy-rules	Нет
FIREWALL_REDIS_SETTINGS	str	Стрим настроек в Redis	app:conf	Нет
FIREWALL_REGISTRATION_TOKEN	str	Токен регистрации для файрвола		Да
FIREWALL_SERVER_CERTS	str	Папка сертификатов	data/certs	Нет
FIREWALL_SERVER_FROM	str	Адрес сервера	0.0.0.0:9991	Нет
FIREWALL_SERVER_GITHUB	str	Папка артефактов GitHub	data/artifacts/github	Нет
FIREWALL_SERVER_IMAGE	str	Папка артефактов образов	data/artifacts/images	Нет
FIREWALL_SERVER_NPM	str	Папка артефактов NPM	data/artifacts/npm	Нет
FIREWALL_SERVER_OSA	str	Папка артефактов OSA	data/osa	Нет
FIREWALL_SERVER_OSA_CONF_PATH	str	Конфигурационный файл OSA	osa_conf.json	Нет
FIREWALL_SERVER_PHP	str	Папка артефактов PHP	data/artifacts/php	Нет
FIREWALL_SERVER_PYPI	str	Папка артефактов PyPI	data/artifacts/pypi	Нет
FIREWALL_SERVER_TIMEOUT	str	Таймаут соединений с сервером	600s	Нет
HUB_REPORT_ENDPOINT_URL	str	Интеграция с AppSecHub	development	Нет
HUB_TOKEN	str	Токен для интеграции с AppSecHub	development	Нет
OIDC_RP_CLIENT_ID	str	Вход по SSO	user1	Нет
OIDC_RP_SIGN_ALGO	str	Алгоритм SSO	RS256	Нет
REDIS_DB	int	Номер БД Redis	1	Нет
REDIS_HOST	str	Хост Redis	redis	Нет

Переменная	Тип	Описание	Значение по умолчанию	Обязательная
REDIS_KEY_POLICY_RULES	str	Ключ для политик из файрвола	app:firewall-policy-rules	Нет
REDIS_PORT	int	Порт Redis	6379	Нет
REDIS_STREAM_CONF	str	Стрим конфигурации в Redis	app:conf	Нет
REDIS_STREAM_FIREWALL	str	Стрим firewall activity пакетов	app:packets	Нет
REDIS_STREAM_FIREWALL_BACKUP	str	Стрим бекапа firewall activity	app:packetsbackup	Нет
REDIS_STREAM_GITHUB	str	Стрим GitHub скоринга	app:scoring_out	Нет
REDIS_STREAM_IDENTITY	str	Стрим identities	app:identities	Нет
REDIS_STREAM_POLICY	str	Стрим политик	app:policies	Нет
SECRET_KEY	str	Секретный ключ Django	secret	Нет
TOKEN_EXPIRE_MINUTES	int	Время истечения токена (минуты)	100000	Нет

3. Развертывание Dependency Firewall через Helm-чарт

Приложение состоит из следующих компонентов

1. backend (см. п. 3.1)
2. feeder-service (см. п. 3.2)
3. firewall (см. п. 3.3)
4. frontend (см. п. 3.4)
5. git-scoring (см. п. 3.5)

В данном разделе описано развертывание приложения в kubernetes с помощью helm при использовании universal-chart (файл universal-chart.tgz приложен к настоящей инструкции):

1. kubernetes >= 1.28
2. helm = 3

Чарт гибко конфигурируемый, в данном документе описаны возможные параметры конфигурации (см. п. 3.6).

Приложения конфигурируются через переменные окружения, описанные в разделе Передача секретов и параметров (ссылка на п. 3.8).

Инфраструктурные сервисы необходимые для развертывания:

1. PostgreSQL
2. Redis
3. MinIO

3.1 Развёртывание компонента backend через Helm-чарт

Команда для установки компонента backend:

```
helm upgrade -i backend . -f values.yaml
```

Приложению требуется доступ к:

1. Redis service (учетная запись с rw правами).
2. PostgreSQL service (учетная запись с rw правами).
3. Minio (s3 бакет с rw правами).

Сервисы:

1. backend
2. backend-consumer
3. github-consumer
4. celery
5. celery-beat

Пример развертывания компонента backend:

```
deployments:
  backend:
    replicas: 2
    extraSelectorLabels:
      app: backend-api
    automountServiceAccountToken: false
    securityContext:
      fsGroup: 1000
      runAsGroup: 1000
      runAsNonRoot: true
      runAsUser: 1000
      seccompProfile:
        type: RuntimeDefault
    initContainers:
    - name: backend-migrate
      command: ["sh", "-c", "python manage.py migrate"]
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop:
          - ALL
        privileged: false
        readOnlyRootFilesystem: false
      envFrom:
      - env
    containers:
    - name: backend
      command: ["uwsgi", "--ini", "/code/src/uwsgi.ini", "--http", "0.0.0.0:8000"]
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop:
          - ALL
        privileged: false
        readOnlyRootFilesystem: false
      resources:
        limits:
          cpu: 1
          memory: 4Gi
        requests:
          cpu: 200m
          memory: 300Mi
      envFrom:
      - env
      ports:
      - name: http
```

```
      containerPort: 8000
backend-consumer:
  automountServiceAccountToken: false
  securityContext:
    fsGroup: 1000
    runAsGroup: 1000
    runAsNonRoot: true
    runAsUser: 1000
    seccompProfile:
      type: RuntimeDefault
  containers:
  - name: backend-consumer
    command: ["sh", "-c", "python manage.py consumer"]
    securityContext:
      allowPrivilegeEscalation: false
      capabilities:
        drop:
        - ALL
      privileged: false
      readOnlyRootFilesystem: false
    resources:
      limits:
        cpu: 300m
        memory: 4Gi
      requests:
        cpu: 200m
        memory: 2Gi
    envFrom:
    - env
celery-beat:
  automountServiceAccountToken: false
  securityContext:
    fsGroup: 1000
    runAsGroup: 1000
    runAsNonRoot: true
    runAsUser: 1000
    seccompProfile:
      type: RuntimeDefault
  containers:
  - name: celery-beat
    command: ["celery", "--app=celery_app", "beat"]
    securityContext:
      allowPrivilegeEscalation: false
      capabilities:
        drop:
        - ALL
      privileged: false
      readOnlyRootFilesystem: false
    resources:
      limits:
        cpu: 200m
        memory: 300Mi
      requests:
        cpu: 100m
        memory: 300Mi
    envFrom:
    - env
celery:
  automountServiceAccountToken: false
  securityContext:
    fsGroup: 1000
```

```
      runAsGroup: 1000
      runAsNonRoot: true
      runAsUser: 1000
      seccompProfile:
        type: RuntimeDefault
    containers:
    - name: celery
      command: ["celery", "--app=celery_app", "worker", "--loglevel=info", "--
concurrency=4"]
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop:
          - ALL
        privileged: false
        readOnlyRootFilesystem: false
      resources:
        limits:
          cpu: 500m
          memory: 4Gi
        requests:
          cpu: 500m
          memory: 4Gi
      envFrom:
      - env
  github-consumer:
    automountServiceAccountToken: false
    securityContext:
      fsGroup: 1000
      runAsGroup: 1000
      runAsNonRoot: true
      runAsUser: 1000
      seccompProfile:
        type: RuntimeDefault
    containers:
    - name: github-consumer
      command: ["python", "manage.py", "github_consumer"]
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop:
          - ALL
        privileged: false
        readOnlyRootFilesystem: false
      resources:
        limits:
          cpu: 300m
          memory: 500Mi
        requests:
          cpu: 200m
          memory: 500Mi
      envFrom:
      - env
```

Необходимо открыть эндпоинты (Ingresses):

1. /api
2. /admin
3. /swagger

4. /backend-static

Пример:

```
ingresses:
  backend:
    annotations:
      nginx.ingress.kubernetes.io/use-regex: "true"
      nginx.ingress.kubernetes.io/rewrite-target: /api/$2
    ingressClassName: nginx-internal
    hosts:
    - hostname: df.your.domain.com
      paths:
      - serviceName: backend
        servicePort: http
        path: /api(/|$)(.*)
    tlsName: df
  services:
    ingressClassName: nginx-internal
    hosts:
    - hostname: df.your.domain.com
      paths:
      - serviceName: backend
        servicePort: http
        path: /swagger/
      - serviceName: backend
        servicePort: http
        path: /backend-static/
      - serviceName: backend
        servicePort: http
        path: /admin/
    tlsName: df
```

3.2 Развёртывание компонента feeder-service через Helm-чарт

Тип установки: StatefulSet

Команда для установки компонента feeder-service:

```
helm upgrade -i feeder-service . -f values.yaml
```

Приложению требуется доступ к:

1. Redis service
2. CICADA8 Registry (опционально)

Приложение развёртывается как statefulSet и хранит свои данные на диске.

Тома (volumes) представлены в таблице 4.

Таблица 4 – Тома (volumes) компонента feeder-service

Имя тома	Точка монтирования	Рекомендуемый размер	Комментарий
feeder-tmp	/tmp	2G	PV

Пример развертывания StatefulSets:

```
statefulSets:
```

```
  api:
    automountServiceAccountToken: false
    securityContext:
      fsGroup: 1000
      runAsGroup: 1000
      runAsNonRoot: true
      runAsUser: 1000
      seccompProfile:
        type: RuntimeDefault
    containers:
    - name: api
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop:
          - ALL
        privileged: false
        readOnlyRootFilesystem: false
      resources:
        limits:
          cpu: 200m
          memory: 256Mi
        requests:
          cpu: 100m
          memory: 128Mi
      envFrom:
      - env
      ports:
      - name: http
        containerPort: 8089
      volumeMounts:
      - mountPath: "/tmp"
        name: feeder-tmp
    volumeClaimTemplates:
    - metadata:
        name: feeder-tmp
      spec:
        accessModes: ["ReadWriteOnce"]
        storageClassName: "yourstorageclass"
        resources:
          requests:
            storage: 2G
```

Входные точки (Ingresses):

```
ingresses:
  default:
    ingressClassName: nginx
    hosts:
    - hostname: df-feeder.your.domain.con
      paths:
      - serviceName: feeder-service
        servicePort: http
```

3.3 Развёртывание компонента firewall через Helm-чарт

Команда для установки компонента firewall:

```
helm upgrade -i firewall . -f values.yaml
```

Приложению требуется доступ к:

1. Redis service
2. Feeder service
3. Backend

Для работы приложения требуется открыть tcp порт, определенный в переменной FIREWALL_SERVER_FROM.

Возможные варианты реализации:

1. LoadBalancer

Определить в values.yaml:

```
  services:
    type: LoadBalancer
```

В зависимости от реализации (metallb, cilium) добавить необходимые annotations, labels.

2. Ingress Controller

В данном случае требуется на ingress controller открыть дополнительный tcp порт и пробросить его на приложение. Протестировано на ingress-nginx и haproxy.

Также требуются 80, 443 порты для работы proxy.

Приложение развертывается как statefulSet и хранит свои данные на диске.

Тома (volumes) представлены в таблице 5.

Таблица 5 – Тома (volumes) компонента firewall

Имя тома	Точка монтирования	Рекомендуемый размер	Комментарий
firewall-data	/data/osa	10G	PV
firewall-tmp	/data/artifacts	50G	PV
osa-tmp	/tmp	30G	PV
certs-tmp	/data/certs	50Mi	emptyDir
firewall-registration	/data/registration	10Mi	PV

Пример развертывания StatefulSets:

```
statefulSetsGeneral:
  extraVolumes:
  - name: certs-tmp
    emptyDir:
      sizeLimit: 50Mi
statefulSets:
  firewall:
    replicas: 2
    automountServiceAccountToken: false
    securityContext:
```

```
    fsGroup: 1000
    runAsGroup: 1000
    runAsNonRoot: true
    runAsUser: 1000
    seccompProfile:
      type: RuntimeDefault
  containers:
  - name: firewall
    securityContext:
      allowPrivilegeEscalation: false
      capabilities:
        drop:
        - ALL
      privileged: false
      readOnlyRootFilesystem: false
    resources:
      limits:
        cpu: '6'
        memory: 12Gi
      requests:
        cpu: '4'
        memory: 4Gi
    envFrom:
    - env
    ports:
    - name: http
      containerPort: 9991
    volumeMounts:
    - mountPath: "data/osa"
      name: firewall-data
    - mountPath: "/data/certs"
      name: certs-tmp
    - mountPath: "/data/artifacts"
      name: firewall-tmp
    - mountPath: "/tmp"
      name: osa-tmp
    - mountPath: /data/registration
      name: firewall-registration
  volumeClaimTemplates:
  - metadata:
      name: firewall-tmp
    spec:
      accessModes: ["ReadWriteOnce"]
      storageClassName: "vcd-disk-ultra"
      resources:
        requests:
          storage: 100G
  - metadata:
      name: osa-tmp
    spec:
      accessModes: ["ReadWriteOnce"]
      storageClassName: "vcd-disk-fast"
      resources:
        requests:
          storage: 30G
  - metadata:
      name: firewall-data
    spec:
      accessModes: ["ReadWriteOnce"]
      storageClassName: "vcd-disk-ultra"
      resources:
```

```
          requests:
            storage: 10G
    - metadata:
        name: firewall-registration
      spec:
        accessModes: ["ReadWriteOnce"]
        storageClassName: "vcd-disk-basic"
        resources:
          requests:
            storage: 10Mi
```

Входные точки (Ingresses):

```
ingresses:
  default:
    ingressClassName: nginx
    hosts:
    - hostname: df-proxy.your.domain.con
      paths:
      - serviceName: firewall
        servicePort: http
```

3.4 Развёртывание компонента frontend через Helm-чарт

Команда для установки компонента frontend:

```
helm upgrade -i backend . -f values.yaml
```

Пример развертывания:

```
deployments:
  frontend:
    automountServiceAccountToken: false
    securityContext:
      fsGroup: 65532
      runAsGroup: 65532
      runAsNonRoot: true
      runAsUser: 65532
      seccompProfile:
        type: RuntimeDefault
    containers:
    - name: frontend
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop:
          - ALL
        privileged: false
        readOnlyRootFilesystem: true
      resources:
        limits:
          cpu: 200
          memory: 256Mi
        requests:
          cpu: 100m
          memory: 128Mi
      ports:
      - name: http
        containerPort: 8080
      volumeMounts:
```

```
      - mountPath: /var/lib/nginx/tmp
        name: nginx-tmp
      - mountPath: /var/run
        name: run-tmp
    volumes:
    - type: emptyDir
      name: nginx-tmp
    - type: emptyDir
      name: run-tmp
```

Необходимо открыть эндпоинты (Ingresses):

1. /api
2. /admin
3. /swagger
4. /backend-static

Пример:

```
ingresses:
  default:
    ingressClassName: nginx-internal
    hosts:
    - hostname: df.your.domain.com
      paths:
      - serviceName: frontend
        servicePort: http
```

3.5 Развёртывание компонента git-scoring через Helm-чарт

Команда для установки компонента git-scoring:

```
helm upgrade -i git-scoring . -f values.yaml
```

Приложению требуется доступ к Git.

Сервисы: git-scoring

Пример развертывания:

```
deploymentsGeneral:
  extraVolumes:
  - name: tmp
    emptyDir:
      sizeLimit: 512Mi
deployments:
  api:
    automountServiceAccountToken: false
    securityContext:
      fsGroup: 1000
      runAsGroup: 1000
      runAsNonRoot: true
      runAsUser: 1000
      seccompProfile:
        type: RuntimeDefault
    containers:
    - name: api
```

```
      securityContext:
        allowPrivilegeEscalation: false
        capabilities:
          drop:
          - ALL
        privileged: false
        readOnlyRootFilesystem: false
      resources:
        limits:
          cpu: 200m
          memory: 256Mi
        requests:
          cpu: 100m
          memory: 128Mi
      envFrom:
      - env
      volumeMounts:
      - mountPath: "/tmp"
        name: tmp
```

3.6 Параметры конфигурации для развёртывания приложения через Helm-чарт

Для развертывания приложения в kubernetes используется чарт Universal chart (файл universal-chart.tgz приложен к настоящей инструкции).

Параметры Global представлены в таблице 6.

Таблица 6 – Параметры Global

Название	Описание	Значение
global.kubeVersion	Глобальное переопределение версии Kubernetes	`""`

Параметры Generic представлены в таблице 7.

Таблица 7 – Параметры Generic

Название	Описание	Значение
generic.labels	Метки для добавления ко всем развернутым объектам	`{}`
generic.annotations	Аннотации для добавления ко всем развернутым объектам	`{}`
generic.extraSelectorLabels	SelectorLabels для добавления в развертывания и службы	`{}`
generic.podLabels	Метки для добавления ко всем развернутым подам	`{}`
generic.podAnnotations	Аннотации для добавления ко всем развернутым модулям	`{}`
generic.serviceAccountName	Имя ServiceAccount, используемое в зависимости от рабочей нагрузки	`""`
generic.hostAliases	Алиасы узлов Pods, используемые в зависимости от рабочей нагрузки	`[]`

Название	Описание	Значение
generic.dnsPolicy	dnsPolicy для подов рабочей нагрузки	`""`
generic.volumes	Массив типизированных томов (Volumes) для добавления ко всем развернутым рабочим нагрузкам	`[]`
generic.volumeMounts	Массив k8s VolumeMounts для добавления ко всем развернутым рабочим нагрузкам	`[]`
generic.extraVolumes	Массив k8s Volumes для добавления ко всем развернутым рабочим нагрузкам	`[]`
generic.extraImagePullSecrets	Множество существующих PullSecrets для добавления ко всем развернутым рабочим нагрузкам	`[]`
generic.usePredefinedAffinity	Использование Affinity presets во всех рабочих нагрузках по умолчанию	`true`
generic.extSecrets	Блокирование использования внешних секретов	`{}`

Чарт поддерживает управление секретами через external secrets operator. Требуется указать secretStoreName. Для каждой записи в secrets будет создан секрет, данные для которого будут взяты из vault kv по пути, указаном в path. Секрет будет создан с именем <release-name>-".secrets.<name>".

Пример:

```
extSecrets:
  enabled: true
  secretStoreName: "secretstore"
  secrets:
    env:
      path: firewall/dev/cd/env
```

Параметры Common представлены в таблице 8.

Таблица 8 – Параметры Common

Название	Описание	Значение
kubeVersion	Переопределение версии Kubernetes	`""`
nameOverride	Строка для переопределения названия релиза	`""`
envs	Карта переменных среды, которая будет развернута как	`{}`

Название	Описание	Значение
	ConfigMap с именем RELEASE_NAME-env	
envsString	Строка с отображением переменных среды, которые будут развернуты в виде ConfigMap с именем RELEASE_NAME-envs	`""`
secretEnvs	Отображение переменных среды, которые будут развернуты как Secret с именем RELEASE_NAME-secret-envs	`{}`
secretEnvsString	Строка с отображением переменных среды, которые будут развернуты как секреты, с именем RELEASE_NAME-secret-envs	`""`
imagePullSecrets	Карта секретов registry в формате .dockerconfigjson	`{}`
defaultImage	Образ Docker, который будет использоваться по умолчанию	`[]`
defaultImageTag	Тег образа Docker, который будет использоваться по умолчанию	`[]`
defaultImagePullPolicy	Политика извлечения образов Docker, которая будет использоваться по умолчанию	`"IfNotPresent"`
podAffinityPreset	Предустановленное соответствие пода. Игнорируется, если задано соответствие рабочей нагрузке. Допустимые значения: soft or hard.	`soft`
podAntiAffinityPreset	Предустановка anti-affinity пода. Игнорируется, если установлено соответствие рабочей нагрузки. Допустимые значения: soft or hard	`soft`
nodeAffinityPreset.type	Заданный тип привязки к ноде. Игнорируется, если установлена привязка к рабочей нагрузке. Допустимые значения: soft or hard	`""`
nodeAffinityPreset.key	Соответствующий ключ метки ноды. Игнорируется, если установлено соответствие рабочей нагрузки	`""`

Название	Описание	Значение
nodeAffinityPreset.values	Node label values to match. Ignored if workload affinity is set	`[]`
extraDeploy	Map of extra objects (k8s manifests or Helm templates) to deploy with the release.	`[]`
diagnosticMode.enabled	Включение режима диагностики (все датчики будут отключены, а команда переопределена)	`false`
diagnosticMode.command	Команда для переопределения всех контейнеров при развертывании	`["sleep"]`
diagnosticMode.args	Аргументы для переопределения всех контейнеров в развертывании	`["infinity"]`
releasePrefix	Переопределение префикса для всех имен манифестов. Имя релиза используется по умолчанию. Следует использовать "-", чтобы сделать его пустым.	`""`

ingresses — это карта параметров Ingress, где key — имя хоста (домен) Ingress.

Параметры Ingress представлены в таблице 9.

Таблица 9 – Параметры Ingress

Название	Описание	Значение
name	Пользовательское имя входящего трафика, если будет использоваться пустое имя хоста входящего трафика	`""`
labels	Дополнительные метки для входа	`{}`
annotations	Дополнительные аннотации для входа	`{}`
certManager.issuerName	Имя эмитента CertManager для входящего TLS	`""`
certManager.issuerType	Тип эмитента CertManager для входящего TLS	`"cluster-issuer"`
ingressClassName	Имя ingressClass	`""`
tlsName	Имя секрета, используемого для генерации TLS Cert Manager	`""`
hosts	Массив объектов входящих хостов	`[]`
extraTls	Массив входящих параметров TLS.	`[]`

Параметры Ingress hosts object представлены в таблице 10.

Таблица 10 – Параметры Ingress hosts object

Название	Описание	Значение

hostname	Имя хоста для обслуживания входящего трафика, если будет использоваться пустое имя входящего хоста	`""`
paths	Массив объектов входящих путей	`[]`

Параметры Ingress paths object представлены в таблице 11.

Таблица 11 – Параметры Ingress paths object

Название	Описание	Значение
path	URL-путь	`"/"`
pathType	Тип входного пути	`"Prefix"`
serviceName	Название сервиса маршрутизации запросов	`""`
servicePort	Имя или номер сервисного порта для маршрутизации запросов	`""`

Параметры Services представлены в таблице 12. Services — это карта параметров сервисов, где key — имя сервиса.

Таблица 12 – Параметры Services

Название	Описание	Значение
labels	Дополнительные метки для сервиса	`{}`
annotations	Дополнительные аннотации для сервиса	`{}`
type	Тип сервиса	`""`
loadBalancerIP	IP сервиса с типом LoadBalancer	`""`
loadBalancerSourceRanges	Источники балансировщика нагрузки сервиса	`[]`
externalTrafficPolicy	Политика внешнего трафика сервиса	`"Cluster"`
healthCheckNodePort	Порт узла проверки работоспособности (числовой номер порта) для сервиса	`` ``
externalIPs	Массив внешних IP-адресов, которые маршрутизируют к одному или нескольким нодам кластера	`[]`
ports	Массив объектов сервисного порта	`[]`
extraSelectorLabels	Дополнительные selectorLabels для выбранной рабочей нагрузки	`{}`
clusterIP	Значение параметра clusterIP сервиса	`""`

Параметры Service ports object представлены в таблице 13.

Таблица 13 – Параметры Service ports object

Название	Описание	Значение
name	Название сервисного порта	`""`

Название	Описание	Значение
protocol	Протокол сервисного порта	`"TCP"`
port	Номер порта сервиса	``
targetPort	Номер целевого порта сервиса	``
nodePort	Номер NodePort сервиса	``

Параметры deploysGeneral представлены в таблице 14. deploysGeneral — это карта параметров развертываний, которая используется для всех развертываний.

Таблица 14 – Параметры deploysGeneral

Название	Описание	Значение
deploymentsGeneral.labels	Метки для добавления ко всем развертываниям	`{}`
deploymentsGeneral.annotations	Аннотации для добавления ко всем развертываниям	`{}`
deploymentsGeneral.envsFromConfigmap	Карта ConfigMaps и ее envs	`{}`
deploymentsGeneral.envsFromSecret	Карта Секретов и ее envs	`{}`
deploymentsGeneral.env	Массив дополнительных переменных окружения	`[]`
deploymentsGeneral.envConfigmaps	Массив имен Configmaps с дополнительными envs	`[]`
deploymentsGeneral.envSecrets	Массив имен секретов с дополнительными envs	`[]`
deploymentsGeneral.envFrom	Массив дополнительных объектов envFrom	`[]`
deploymentsGeneral.extraVolumes	Массив k8s Volumes для добавления ко всем развертываниям	`[]`
deploymentsGeneral.volumeMounts	Массив k8s VolumeMounts для добавления ко всем развертываниям	`[]`
deploymentsGeneral.usePredefinedAffinity	Использовать предустановки Affinity во всех развертываниях по умолчанию	`false`

Параметры deployments представлены в таблице 15. Deployments — это карта параметров Deployment, где key — имя Deployment.

Таблица 15 – Параметры deployments

Название	Описание	Значение
labels	Дополнительные метки для развертывания	`{}`

Название	Описание	Значение
annotations	Дополнительные аннотации для развертывания	`{}`
replicas	Количество реплик развертывания	`1`
strategy	Стратегия развертывания	`{}`
extraSelectorLabels	Дополнительные метки селектора для развертывания	`{}`
podLabels	Дополнительные метки подов для развертывания	`{}`
podAnnotations	Дополнительные аннотации подов для развертывания	`{}`
serviceAccountName	Имя ServiceAccount, используемое при развертывании	`""`
hostAliases	Алиасы хостов Pods	`[]`
affinity	Affinity для развертывания; назначение реплик подов	`{}`
securityContext	Контекст безопасности для развертывания подов	`{}`
dnsPolicy	DnsPolicy для развертывания подов	`""`
nodeSelector	Метки нод для развертывания; назначение подов	`{}`
tolerations	Допуски для развертывания; назначение реплик подов	`[]`
imagePullSecrets	УСТАРЕЛО. Массив существующих секретов pull	`[]`
extraImagePullSecrets	Массив существующих секретов pull	`[]`
terminationGracePeriodSeconds	Целое число, устанавливающее период завершения для подов	`30`
initContainers	Массив initContainers развертывания (container objects)	`[]`
containers	Массив контейнеров развертывания (container objects)	`[]`
volumes	Массив объектов volumes, типизированных для развертывания	`[]`
extraVolumes	Массив k8s Volumes для добавления в развертывания	`[]`

Параметры statefulSetsGeneral представлены в таблице 16. statefulSetsGeneral — это карта параметров StatefulSets, которая используется для всех StatefulSets.

Таблица 16 – Параметры statefulSetsGeneral

Название	Описание	Значение
statefulSetsGeneral.labels	Метки для добавления во все StatefulSets	`{}`
statefulSetsGeneral.annotations	Аннотации для добавления ко всем StatefulSets	`{}`
statefulSetsGeneral.envsFromConfigmap	Карта ConfigMaps и ее envs	`{}`
statefulSetsGeneral.envsFromSecret	Карта Секретов и ее envs	`{}`
statefulSetsGeneral.env	Массив дополнительных переменных окружения	`[]`
statefulSetsGeneral.envConfigmaps	Массив имен Configmaps с дополнительными envs	`[]`
statefulSetsGeneral.envSecrets	Массив имен секретов с дополнительными envs	`[]`
statefulSetsGeneral.envFrom	Массив дополнительных объектов envFrom	`[]`
statefulSetsGeneral.extraVolumes	Массив k8s Volumes для добавления во все StatefulSets	`[]`
statefulSetsGeneral.volumeMounts	Массив k8s VolumeMounts для добавления во все StatefulSets	`[]`
statefulSetsGeneral.usePredefinedAffinity	Использовать предустановки Affinity во всех StatefulSets по умолчанию	`false`

Параметры statefulSets представлены в таблице 17. statefulSets — это карта параметров StatefulSets, где key — имя StatefulSets.

Таблица 17 – Параметры statefulSets

Название	Описание	Значение
labels	Дополнительные метки для statefulSet	`{}`
annotations	Дополнительные аннотации для statefulSet	`{}`
replicas	Количество реплик StatefulSet	`1`
minReadySeconds	StatefulSet minReadySeconds	``
strategy	Стратегия StatefulSet	`{}`
extraSelectorLabels	Дополнительные метки селектора для statefulSet	`{}`
podLabels	Дополнительные метки подов для statefulSet	`{}`
podAnnotations	Дополнительные аннотации подов для statefulSet	`{}`

Название	Описание	Значение
serviceAccountName	Имя ServiceAccount, которое будет использоваться statefulSet	`""`
hostAliases	Алиасы хостов подов	`[]`
affinity	Affinity statefulSet; назначение реплик подов	`{}`
securityContext	Контекст безопасности для подов statefulSet	`{}`
dnsPolicy	DnsPolicy для подов statefulSet	`""`
nodeSelector	Метки нод для statefulSet; назначение подов	`{}`
tolerations	Допуски для statefulSet; назначение реплик подов	`[]`
imagePullSecrets	УСТАРЕЛО. Массив существующих секретов pull	`[]`
extraImagePullSecrets	Массив существующих секретов pull	`[]`
terminationGracePeriodSeconds	Целое число, устанавливающее период завершения для подов	`30`
initContainers	Массив statefulSet initContainers (container objects)	`[]`
containers	Массив контейнеров statefulSet (container objects)	`[]`
volumes	Массив типизированных объектов volume statefulSet	`[]`
extraVolumes	Массив k8s Volumes для добавления в statefulSets	`[]`
volumeClaimTemplates	Массив k8s volumeClaimTemplates для добавления в statefulSets	`[]`

Параметры Container object представлены в таблице 18.

Таблица 18 – Параметры Container object

Название	Описание	Значение
name	Название контейнера	`""`
image	Docker-образ контейнера	`""`
imageTag	Тег образа Docker контейнера	`"latest"`
imagePullPolicy	Политика извлечения образов Docker	`"IfNotPresent"`
securityContext	Контекст безопасности для контейнера	`{}`

Название	Описание	Значение
command	Переопределение команды контейнера (список или строка)	`[]`
commandMaxDuration	Продолжительность выполнения команды (только для jobs и cronJobs)	``
commandDurationAlert	Создать оповещение Prometheus о превышении времени выполнения команды (только для jobs и cronJobs)	``
args	Переопределение аргументов контейнера	`[]`
envsFromConfigmap	Карта ConfigMaps and её envs	`{}`
envsFromSecret	Карта Секретов и её envs	`{}`
env	Массив дополнительных переменных окружения	`[]`
envConfigmaps	Массив имен Configmaps с дополнительными envs	`[]`
envSecrets	Массив имен секретов с дополнительными envs	`[]`
envFrom	Массив дополнительных объектов envFrom	`[]`
ports	Массив портов, которые будут доступны из контейнера	`[]`
lifecycle	Хуки жизненного цикла контейнеров	`{}`
livenessProbe	Объект проверки жизнеспособности контейнера	`{}`
readinessProbe	Объект проверки готовности к использованию контейнера	`{}`
resources	Запросы ресурсов и ограничения для контейнера	`{}`
volumeMounts	Массив k8s Volume mounts	`[]`

Параметры Secrets представлены в таблице 19. secrets — это карта параметров Secret, где key — имя Secret.

Таблица 19 – Параметры Secrets

Название	Описание	Значение
type	Тип секрета	`"Opaque"`
labels	Дополнительные метки секретов	`{}`
annotations	Дополнительные аннотации секретов	`{}`
data	Карта данных секретов	`{}`

Объект Secret data — это карта, где значение может быть строкой, строкой в кодировке JSON или base64 с префиксом `b64:`.

Параметры ConfigMaps представлены в таблице 20. configMaps — это карта параметров ConfigMap, где key — имя ConfigMap.

Таблица 20 – Параметры ConfigMaps

Название	Описание	Значение
labels	Дополнительные метки ConfigMap	`{}`
annotations	Дополнительные аннотации ConfigMap	`{}`
data	Карта данных ConfigMap	`{}`

Параметры PersistentVolumeClaims представлены в таблице 21. pvcs — это карта параметров PersistentVolumeClaim, где key — имя PersistentVolumeClaim.

Таблица 21 – Параметры ConfigMaps

Название	Описание	Значение
labels	Метки Extra Persistent Volume Claim	`{}`
annotations	Дополнительные аннотации Persistent Volume Claim	`{}`
accessModes	Режимы доступа к Persistent Volume	`[]`
volumeMode	Persistent Volume volume mode	`"Filesystem"`
storageClassName	Имя Persistent Volume Storage Class	`""`
selector	Селектор меток для дальнейшей фильтрации набора volumes	`{}`

Параметры typed Volumes представлены в таблице 22.

Таблица 22 – Параметры typed Volumes

Название	Описание	Значение
type	Тип ресурса volume ("configMap","secret","pvc")	`""`
name	Имя ресурса, который будет использоваться с префиксом релиза	`""`
originalName	Оригинальное название ресурса	`""`
items	Массив элементов volume	`[]`

Параметры cronJobsGeneral представлены в таблице 23. cronJobsGeneral — это карта параметров CronJobs, которая используется для всех CronJobs.

Таблица 23 – Параметры cronJobsGeneral

Название	Описание	Значение
cronJobsGeneral.labels	Дополнительные метки для всех CronJobs	`{}`
cronJobsGeneral.annotations	Дополнительные аннотации для всех CronJobs	`{}`
cronJobsGeneral.envsFromConfigmap	Карта ConfigMaps и ее envs	`{}`
cronJobsGeneral.envsFromSecret	Карта Secrets и ее envs	`{}`
cronJobsGeneral.env	Массив дополнительных переменных окружения	`[]`
cronJobsGeneral.envConfigmaps	Массив имен Configmaps с дополнительными envs	`[]`
cronJobsGeneral.envSecrets	Массив имен секретов с дополнительными envs	`[]`
cronJobsGeneral.envFrom	Массив дополнительных объектов envFrom	`[]`
cronJobsGeneral.startingDeadlineSeconds	Продолжительность запуска всех CronJobs (игнорируется, если определено на уровне CronJob)	``
cronJobsGeneral.successfulJobsHistoryLimit	Ограничение на хранение количества выполненных Job (игнорируется, если определено на уровне CronJob)	`3`
cronJobsGeneral.failedJobsHistoryLimit	Ограничение на хранение количества невыполненных Job (игнорируется, если определено на уровне CronJob)	`1`
cronJobsGeneral.parallelism	Сколько подов Job могут быть запущены параллельно (игнорируется, если определено на уровне CronJob)	`1`
cronJobsGeneral.completions	Сколько подов необходимо завершить для завершения Job (игнорируется, если определено на уровне CronJob)	`1`

Название	Описание	Значение
cronJobsGeneral.activeDeadlineSeconds	Продолжительность Job (игнорируется, если определено на уровне CronJob)	`100`
cronJobsGeneral.backoffLimit	Количество повторных попыток, прежде чем Job будет признано невыполненным (игнорируется, если определено на уровне CronJob)	`6`
cronJobsGeneral.ttlSecondsAfterFinished	TTL для удаления завершенных Jobs (игнорируется, если определено на уровне CronJob)	`100`
cronJobsGeneral.podLabels	Дополнительные метки подов для CronJob (игнорируются, если определены на уровне CronJob)	`{}`
cronJobsGeneral.podAnnotations	Дополнительные аннотации подов для CronJob (игнорируются, если определены на уровне CronJob)	`{}`
cronJobsGeneral.serviceAccountName	Имя ServiceAccount, которую будет использовать Job (игнорируются, если определены на уровне CronJob)	`""`
cronJobsGeneral.hostAliases	Алиасы хостов Pods (игнорируются, если определены на уровне CronJob)	`[]`
cronJobsGeneral.affinity	Affinity для CronJob; назначение реплик подов (игнорируется, если определено на уровне CronJob)	`{}`
cronJobsGeneral.dnsPolicy	DnsPolicy для подов CronJob (игнорируется, если определено на уровне CronJob)	`""`
cronJobsGeneral.extraVolumes	Массив k8s Volumes для добавления во все CronJob	`[]`
cronJobsGeneral.volumeMounts	Массив k8s VolumeMounts для добавления во все CronJobs	`[]`
cronJobsGeneral.usePredefinedAffinity	Использовать предустановки Affinity во всех заданиях CronJob по умолчанию	`false`

Параметры cronJobs представлены в таблице 24. cronJobs — это карта параметров CronJobs, где key — имя CronJob.

Таблица 24 – Параметры cronJobs

Название	Описание	Значение
labels	Дополнительные метки CronJob	`{}`
annotations	Дополнительные аннотации CronJob	`{}`
singleOnly	Запретить политику параллелизма	`"false"`
startingDeadlineSeconds	Продолжительность запуска CronJob	```
successfulJobsHistoryLimit	Ограничение на хранение количества выполненных jobs	`3`
failedJobsHistoryLimit	Ограничение на хранение количества невыполненных jobs	`1`
parallelism	Сколько подов CronJob может работать параллельно	`1`
completions	Сколько подов необходимо завершить для завершения Job	`1`
activeDeadlineSeconds	Подолжительность Job	`100`
backoffLimit	Количество повторных попыток, прежде чем Job будет признано невыполненным	`6`
ttlSecondsAfterFinished	TTL для удаления завершенного CronJob	`100`
podLabels	Дополнительные метки подов для CronJob	`{}`
podAnnotations	Дополнительные аннотации подов для CronJob	`{}`
serviceAccountName	Имя ServiceAccount, используемое CronJob	`""`
hostAliases	Алиасы хостов подов	`[]`

Название	Описание	Значение
affinity	Affinity для CronJob; назначение реплик подов	`{}`
securityContext	Контекст безопасности для подов CronJob	`{}`
dnsPolicy	DnsPolicy для подов CronJob	`""`
nodeSelector	Метки нод для CronJob; назначение подов	`{}`
tolerations	Допуски для CronJob; назначение реплик подов	`[]`
imagePullSecrets	УСТАРЕЛО. Массив существующих секретов pull	`[]`
extraImagePullSecrets	Массив существующих секретов pull	`[]`
initContainers	Массив CronJob initContainers (container objects)	`[]`
containers	Массив контейнеров CronJob (container objects)	`[]`
volumes	Массив volumes, типизированных CronJob	`[]`
extraVolumes	Массив of k8s Volumes для добавления в CronJob	`[]`
restartPolicy	Политика перезапуска Jobs	`"Never"`

Параметры jobsGeneral представлены в таблице 25. jobsGeneral — это карта параметров Jobs, которая используется для всех Jobs.

Таблица 25 – Параметры jobsGeneral

Название	Описание	Значение
jobsGeneral.labels	Дополнительные метки для всех Job	`{}`
jobsGeneral.annotations	Дополнительные аннотации для всех Job	`{}`
jobsGeneral.envsFromConfigmap	Карта ConfigMaps и ее envs	`{}`
jobsGeneral.envsFromSecret	Карта секретови ее envs	`{}`
jobsGeneral.env	Массив дополнительных переменных окружения	`[]`
jobsGeneral.envConfigmaps	Массив имен Configmaps с дополнительными envs	`[]`
jobsGeneral.envSecrets	Массив имен секретов с дополнительными envs	`[]`
jobsGeneral.envFrom	Массив дополнительных объектов envFrom	`[]`
jobsGeneral.parallelism	Сколько Jobs можно выполнять параллельно (игнорируется, если определено на уровне Job)	`1`
jobsGeneral.completions	Сколько подов необходимо завершить для завершения Job (игнорируется, если определено на уровне Job)	`1`
jobsGeneral.activeDeadlineSeconds	Продолжительность Job (игнорируется, если определено на уровне Job)	`100`
jobsGeneral.backoffLimit	Количество повторных попыток, после которых Job будет считаться невыполненным (игнорируется, если определено на уровне Job)	`6`

Название	Описание	Значение
jobsGeneral.ttlSecondsAfterFinished	TTL для удаления завершенного Job (игнорируется, если определено на уровне Job)	`100`
jobsGeneral.podLabels	Дополнительные метки подов для Job (игнорируется, если определено на уровне Job)	`{}`
jobsGeneral.podAnnotations	Дополнительные аннотации подов для Job (игнорируется, если определено на уровне Job)	`{}`
jobsGeneral.serviceAccountName	Имя ServiceAccount, используемое Job (игнорируется, если определено на уровне Job)	`""`
jobsGeneral.hostAliases	Алиасы хостов подов (игнорируется, если определено на уровне Job)	`[]`
jobsGeneral.affinity	Affinity для Job; назначение реплик подов (игнорируется, если определено на уровне Job)	`{}`
jobsGeneral.dnsPolicy	DnsPolicy для подов Job (игнорируется, если определено на уровне Job)	`""`
jobsGeneral.extraVolumes	Массив k8s Volumes для добавления ко всем Jobs	`[]`
jobsGeneral.volumeMounts	Массив k8s VolumeMounts для добавления ко всем Jobs	`[]`
jobsGeneral.usePredefinedAffinity	Использовать предустановки Affinity во всех Jobs по умолчанию	`false`

Параметры jobs представлены в таблице 26. jobs — это карта параметров Jobs, где key — имя Job.

Таблица 26 – Параметры jobs

Название	Описание	Значение
labels	Дополнительные метки для Job	`{}`
annotations	Дополнительные аннотации для Job	`{}`
parallelism	Сколько подов Job могут быть запущены параллельно	`1`
completions	Сколько подов необходимо завершить для завершения Job	`1`
activeDeadlineSeconds	Продолжительность Job	`100`
backoffLimit	Количество повторных попыток, прежде чем Job будет признано невыполненным	`6`
ttlSecondsAfterFinished	TTL для удаления завершенного Hook Job	`100`
podLabels	Дополнительные метки подов для Hook Job	`{}`
podAnnotations	Дополнительные аннотации подов для for Hook Job	`{}`
serviceAccountName	Имя ServiceAccount, используемое при развертывании	`""`
hostAliases	Алиасы хостов подов	`[]`
affinity	Affinity для Hook Job; назначение реплик подов	`{}`
securityContext	Контекст безопасности подов для Hook Job	`{}`
dnsPolicy	DnsPolicy подов для Hook Job	`""`
nodeSelector	Метки нод для Hook Job; назначение подов	`{}`

Название	Описание	Значение
tolerations	Допуски для Hook Job; назначение реплик подов	`[]`
imagePullSecrets	УСТАРЕЛО. Массив существующих секретов pull	`[]`
image.registry	Реестр образов, который будет использоваться по умолчанию	`docker.io`
image.repository	Тег репозитория образов, который будет использоваться по умолчанию	`nginx`
image.tag	Тег образа, который будет использоваться по умолчанию	`latest`
image.pullPolicy	Политика извлечения образов, которая будет использоваться по умолчанию	`"IfNotPresent"`
extraImagePullSecrets	Массив существующих секретов pull	`[]`
initContainers	Массив Hook Job initContainers (container objects)	`[]`
containers	Массив контейнеров Hook Job (container objects)	`[]`
volumes	Массив типизированных volumes Hook Job	`[]`
extraVolumes	Массив k8s Volumes для добавления в Hook Job	`[]`
restartPolicy	Политика перезапуска Job	`"Never"`

Параметры ServiceMonitors представлены в таблице 27. serviceMonitors — это карта параметров Prometheus ServiceMonitor, где key — имя ServiceMonitor.

Таблица 27 – Параметры ServiceMonitors

Название	Описание	Значение
labels	Дополнительные метки ServiceMonitor	{}
endpoints	Массив эндпоинтов ServiceMonitor	[]
extraSelectorLabels	Дополнительные метки ServiceMonitor для выбранной рабочей нагрузки	{}

3.7 Инфраструктурные сервисы для развёртывания приложения через Helm-чарт

Инфраструктурные сервисы можно установить по указанным ниже инструкциям, либо воспользоваться своей инсталяцией/инстансами.

3.7.1 Redis

1. TLDR.

Приложению требуется редис с поддержкой TLS, авторизацией по логину/паролю.

Необходимо опубликовать сервис наружу кластера.

Протестировано на чарте bitnami/redis:

```
helm pull oci://registry-1.docker.io/bitnamicharts/redis
```

2. Дополнительные ресурсы (Additional resources).

Следует создать в кластере ACL file:

```
apiVersion: v1
kind: Secret
metadata:
  name: redis-acl-list
type: Opaque
stringData:
  users.acl: |
    user default on ~* &* +@all >supastronkpass
```

Затем необходимо сгенерировать сертификат для Redis, сохранить в секрет Certificate, key, CA file для использования.

3. Values.

```
architecture: standalone
auth:
  enabled: true
  password: <your pass>
commonConfiguration: |-
  # Enable AOF https://redis.io/topics/persistence#append-only-file
  appendonly yes
  # Disable RDB persistence, AOF persistence already enabled.
  save ""
  aclfile /etc/redis/users.acl
master:
  count: 1
  extraVolumes:
  - name: redis-acl-list
    secret:
      secretName: redis-acl-list
```

```
  extraVolumeMounts:
  - name: redis-acl-list
    readOnly: true
    subPath: users.acl
    mountPath: "/etc/redis/users.acl"
replica:
  replicaCount: 0
tls:
  enabled: true
  authClients: false
  existingSecret: "redis-dev-tls"
  certFilename: "tls.crt"
  certKeyFilename: "tls.key"
  certCAFilename: "ca.crt"
```

3.7.2 Minio

Minio можно установить из чарта:

```
helm pull oci://registry-1.docker.io/bitnamicharts/minio
```

Требуется создать:

1) s3 bucket

2) Policy

```
   {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "AWS": [
                        "*"
                    ]
                },
                "Action": [
                    "s3:GetObject"
                ],
                "Resource": [
                    "arn:aws:s3:::<bucket-name>/media/**",
                    "arn:aws:s3:::<bucket-name>/static/**"
                ]
            }
        ]
    }
```

3) user, сгенерировать service Account, выдать rw-права на bucket.

3.7.3 PostgreSQL

Для бэкенда требуется инстанс БД версия 15, 16.

3.8 Передача секретов и параметров для развёртывания приложения через Helm-чарт

Передать параметры для приложения можно следующими способами:

1. External Secrets.

Пример:

```
extSecrets:
  enabled: true
  secretStoreName: "secretstore"
  secrets:
    env:
      path: firewall/dev/cd/env
statefulSets:
  firewall:
    containers:
      - envFrom:
        - env
```

Значения из vault будут отражены в секрете releasename-env и экспортированы в переменные окружения контейнера.

2. secretEnvs

Создать секрет, экспортировать его содержимое в переменные окружения:

```
statefulSets:
  firewall:
    containers:
      - envFrom:
        - secret-envs
```

3. В разделе Параметры конфигурации для развёртывания приложения через Helm-чарт (см. п. 3.6) представлены еще несколько способов передачи (configmap, direct env, etc..).

4. Развёртывание Dependency Firewall с использованием docker-compose

Приложение состоит из следующих компонентов:

1. backend
2. feeder-service
3. firewall
4. frontend
5. git-scoring

Архив с настройками (depfw.zip) приложен к настоящей инструкции.

Требования к виртуальной машине:

1. CPU: 16.
2. RAM: 32 GB.
3. SSD: 128 GB.
4. Docker.
5. Docker Compose (рекомендуется Compose v2).

Сервисы, необходимые для развёртывания:

1. PostgreSQL

Для бекенда требуется инстанс БД версия 15, 16.

2. Redis

Приложению требуется Redis с поддержкой TLS, авторизацией по логину/паролю.

Следует опубликовать сервис наружу кластера.

3. MinIO

Необходимо создать s3 bucket и Policy, user (сгенерировать service Account, выдать rw права на bucket).

Создание Policy:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "*"
                ]
            },
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket-name>/media/**",
                "arn:aws:s3:::<bucket-name>/static/**"
            ]
        }
    ]
}
```

Требуется открыть для приложения следующие порты:

1. 443/tcp - nginx
2. 9991/tcp - firewall
3. 9000/tcp, 9001/tcp - minio

Сервисы конфигурируются через переменные окружения. Для каждого сервиса предусмотрена переменная .env.

Быстрый старт:

1. Подготовка файлов .env. Заменить все CHANGEME в nginx/default.conf, .backend.env, .feeder.env. Поместить в nginx/tls.crt nginx/tls.key сертификаты для веба.
2. (опционально) Если используется свой УЦ, то подмонтировать в /etc/ssl/certs/ca.crt свой сертификат УЦ.
3. Авторизоваться в registry.cicada8.ru при помощи полученных учетных записей.

```
docker login registry.cicada8.ru
```

4. Запустить контейнеры.

```
docker compose up -d
```

5. Затем необходимо запустить миграции сервиса backend и создать superuser:

```
    docker exec -it depfw-backend-1 bash
    python manage.py migrate
    python manage.py createsuperuser
                                вводим любой валидный email
```

```
                                bypass password validation - yes
    python manage.py shell
           from apps.user.models import User
           user = User.objects.get(id=1)
           user.role = "admin"
           user.save()
           exit
    docker compose restart
```

5. Настройка интеграций

5.1 Интеграция с AppSecHub

Для корректной работы интеграции с сервисом AppSecHub необходимо задать переменные окружения согласно таблице 28.

Таблица 28 – Переменные окружения

Переменная	Описание	Пример значения
HUB_TOKEN	Токен, сформированный на стороне сервиса AppSecHub.	`token`
HUB_REPORT_ENDPOINT_URL	URL для загрузки сформированного отчета в сервис AppSecHub.	`https://<хост appsechub>/hub/rest/integration/report`
APPSECHUB_INTEGRATION_ENABLED	Включение интеграции с AppSecHub.	`true`

5.2 Подключение OIDC провайдера

Для корректной работы SSO-функциональности по протоколу OIDC необходимо:

1. Указать Вашему OpenID Connect провайдеру URL-адрес обратного вызова для Вашего сайта. Эндпоинт для обратного вызова - /api/oidc/callback/.

Примеры URL-адресов обратного вызова представлены в таблице 29.

Таблица 29 – Примеры URL-адресов обратного вызова

URL	Описание
http://127.0.0.1:8000/api/oidc/callback/	URL для локальной разработки.
https://myapp-dev.example.com/api/oidc/callback/	Среда разработки для приложения.
https://myapp.herokuapps.com/api/oidc/callback/	Приложение, запущенное на Heroku.

2. Задать соответствующие переменные окружения.

Переменные окружения OIDC провайдера представлены в таблице 30.

Таблица 30 – Переменные окружения OIDC провайдера

Переменная	Описание
OIDC_RP_CLIENT_ID	client_id, выдается провайдером SSO.
OIDC_RP_CLIENT_SECRET	client_secret, провайдером SSO.
OIDC_RP_SIGN_ALGO	Алгоритм подписи (по умолчанию значение RS256), задается провайдером SSO.

Переменные окружения для эндпоинтов провайдера SSO представлены в таблице 31.

Эти значения относятся к Вашему провайдеру OpenID Connect. Обратитесь к документации провайдера для получения соответствующих значений.

Таблица 31 – Переменные окружения для эндпоинтов провайдера SSO

Переменная	Описание
OIDC_OP_AUTHORIZATION_ENDPOINT	URL провайдера OIDC с информацией об авторизации.
OIDC_OP_TOKEN_ENDPOINT	URL провайдера OIDC с информацией о токене.
OIDC_OP_USER_ENDPOINT	URL провайдера OIDC с информацией о пользователе.
OIDC_OP_JWKS_ENDPOINT	URL провайдера OIDC с информацией о JSON Web Key.
LOGIN_REDIRECT_URL	URL клиентского сервиса (Dependency Firewall), на который происходит переход при успешном входе в систему.
LOGOUT_REDIRECT_URL	URL клиентского сервиса (Dependency Firewall), на который происходит переход при успешном выходе из системы.

ВНИМАНИЕ! Необходимо создать аккаунт в Dependency Firewall с такой же почтой, как и на сервисе провайдера SSO. Если такого аккаунта нет в системе Dependency Firewall, то аутентификация не пройдет.

6. Контакты технических специалистов

Контакты технических специалистов, которые могут проконсультировать по процессу получения доступа к Системе и по вопросам функционирования, представлены в таблице 32.

Таблица 32 – Контакты технических специалистов

Специалист	Должность	Почта
Авраменко Сергей	Руководитель продукта	s.avramenko@cicada8.ru
Щербаков Алексей	Руководитель группы	a.shcherbakov@cicada8.ru