

GIMS MONITORING
РУКОВОДСТВО ПО УСТАНОВКЕ

ВЕРСИЯ 2.1

СОДЕРЖАНИЕ

1	НАЗНАЧЕНИЕ ПРОДУКТА	3
1.1	НАИМЕНОВАНИЕ АС	3
1.2	НАЗНАЧЕНИЕ МОДУЛЕЙ GIMS	3
1.3	ПОДДЕРЖИВАЕМЫЕ ОС	4
2	УСТАНОВКА МОДУЛЕЙ СИСТЕМЫ	5
2.1	ПОДГОТОВКА К УСТАНОВКЕ	5
2.1.1	Подготовка к установке на ALT Linux	5
2.1.2	Подготовка к установке на Red OS	6
2.1.3	Подготовка к установке на Astra Linux	6
2.1.4	Подготовка к установке на Red Hat	7
2.1.5	Подготовка к установке на CentOS	8
2.1.6	Подготовка к установке на Debian	9
2.2	УСТАНОВКА МОДУЛЕЙ GIMS	9
2.2.1	Загрузка образа GIMS из репозитория Gelarm	9
2.2.2	Установка модуля GIMS Portal	11
2.2.3	Установка модуля GIMS Automation	13
2.2.4	Установка модуля GIMS Inventory	16
2.2.5	Установка модуля GIMS Monitoring Fault	18
2.2.6	Установка модуля GIMS Monitoring Performance	20
2.3	НАСТРОЙКА ИНФРАСТРУКТУРЫ КОМПОНЕНТОВ GIMS	21
2.4	КОМАНДЫ ДЛЯ УПРАВЛЕНИЯ УСТАНОВЛЕННЫМИ МОДУЛЯМИ	23
	ПРИЛОЖЕНИЕ А. ПАРАМЕТРЫ НАСТРОЙКИ КОМПОНЕНТ	1

1 НАЗНАЧЕНИЕ ПРОДУКТА

1.1 НАИМЕНОВАНИЕ АС

Полное наименование: Gelarm Infrastructure Management Systems Automation

Обозначение: GIMS или Система

Версии устанавливаемых компонент:

- GIMS Portal v.2.2.4
- GIMS Automation v.2.2.4
- GIMS Inventory v.1.11.0
- GIMS Monitoring Performance v.2.1.34
- GIMS Monitoring Fault v.2.2.3

1.2 НАЗНАЧЕНИЕ МОДУЛЕЙ GIMS

Модуль GIMS Portal - единый web портал для работы с компонентами GIMS.

Модуль GIMS Automation - предназначена для интеграции платформы GIMS с инфраструктурой предприятия. Обеспечивает: сбор инвентарной информации, обработку событий и метрик мониторинга, интеграцию с системами управления инцидентами и проблемами, интеграцию с системами управления плановыми работами, интеграцию с системами учёта активов.

Модуль GIMS Inventory - предназначена для хранения инвентарной информации. Обеспечивает: создание и редактирование моделей данных (классов, атрибутов, взаимосвязей классов), хранение и редактирование данных, поиск, отображение данных на топологии.

Модуль GIMS Monitoring Fault - предназначена для хранения и обработки событий мониторинга. Обеспечивает: дедупликацию событий, корреляцию событий, хранение событий.

Модуль GIMS Monitoring Performance - предназначен для хранения и обработки метрик мониторинга. Обеспечивает: хранение и агрегацию метрик, расчёт пороговых значений.

Подробнее работа каждого из компонент описана в инструкции по администрированию каждого из модулей.

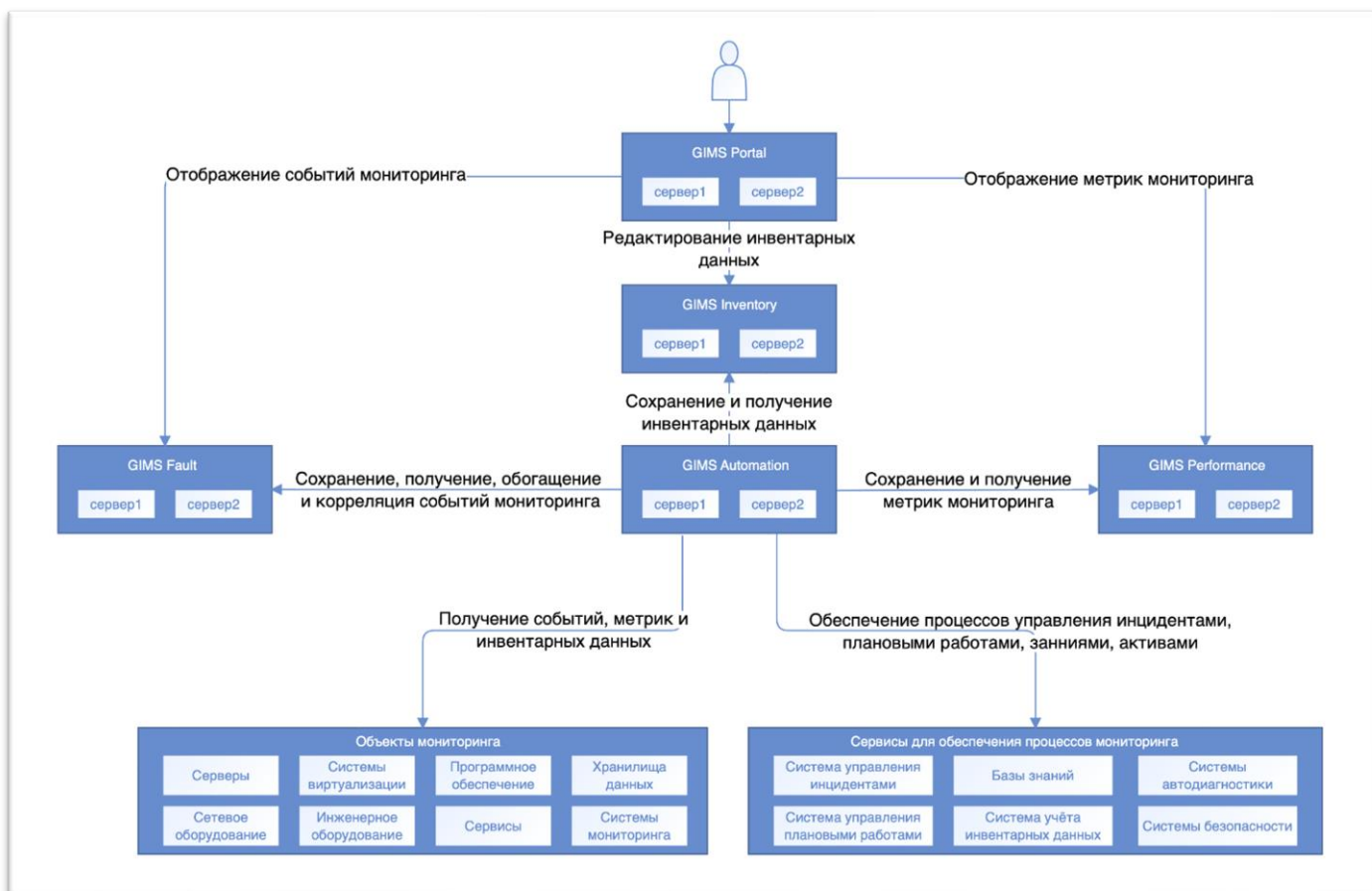


Рисунок 1 - Архитектура решения

Таблица 1 – Максимальное количество создаваемых компонент

НАЗВАНИЕ КОМПОНЕНТА	МАКСИМАЛЬНОЕ КОЛИЧЕСТВО КЛАСТЕРОВ В РАМКАХ ОДНОЙ ИНСТАЛЛЯЦИИ	МАКСИМАЛЬНОЕ КОЛИЧЕСТВО СЕРВЕРОВ В ОДНОМ КЛАСТЕРЕ
GIMS Portal	1	20
GIMS Automation	100	20
GIMS Monitoring Fault	100	20
GIMS Inventory	1	20
GIMS Monitoring Performance	1	20

1.3 ПОДДЕРЖИВАЕМЫЕ ОС

Ниже приведена инструкция для развертывания Системы на следующих ОС.

Таблица 2 – Поддерживаемые ОС

НАЗВАНИЕ ОС	ПОДДЕРЖИВАЕМЫЕ ВЕРСИИ КОМПОНЕНТОВ
ALT Linux	– Docker-ce версии 20 и выше – Docker-compose версии 2 и выше

НАЗВАНИЕ ОС	ПОДДЕРЖИВАЕМЫЕ ВЕРСИИ КОМПОНЕНТОВ
	– Alt Linux 10 и выше
Red OS	– Docker-се версии 20 и выше – Docker-compose версии 2 и выше – Red OS 7.3 и выше
Astra Linux	– Docker-се версии 20 и выше – Docker-compose версии 2 и выше – Astra Linux 1.7 и выше
Red Hat	– Docker-се версии 20 и выше – Docker-compose версии 2 и выше
CentOS	– Docker-се версии 20 и выше – Docker-compose версии 2 и выше
Debian	– Docker-се версии 20 и выше – Docker-compose версии 2 и выше

2 УСТАНОВКА МОДУЛЕЙ СИСТЕМЫ

2.1 ПОДГОТОВКА К УСТАНОВКЕ

2.1.1 Подготовка к установке на ALT Linux

Внимание!!! Шаги подготовки к установке необходимо выполнить на каждом сервере где будут установлены компоненты GIMS.

1. Выполнить синхронизацию времени сервера

Команда: ntpdate pool.ntp.org

2. Установить необходимые пакеты

Команда: apt-get install docker-compose-v2 docker-ce

3. Настроить автоматический запуск сервиса docker

Команда: systemctl enable --now docker

4. Запустить сервис docker

Команда: systemctl start docker

5. Создать пользователя для установки

Команда: useradd gims

6. Предоставить права пользователю на запуск контейнеров

Команда: usermod gims -aG docker

2.1.2 Подготовка к установке на Red OS

Внимание!!! Шаги подготовки к установке необходимо выполнить на каждом сервере где будут установлены компоненты GIMS.

1. Синхронизировать время сервера

Команда: ntpdate pool.ntp.org

2. Установить необходимые пакеты

Команда: yum install docker-compose-v2 docker-ce

3. Настроить автоматический запуск сервиса docker

Команда: systemctl enable --now docker

4. Запустить сервис docker

Команда: systemctl start docker

5. Создать пользователя для установки

Команда: useradd gims

6. Предоставить права пользователю на запуск контейнеров

Команда: usermod gims -aG docker

2.1.3 Подготовка к установке на Astra Linux

Внимание!!! Шаги подготовки к установке необходимо выполнить на каждом сервере где будут установлены компоненты GIMS.

1. Синхронизировать время сервера

Команда: ntpdate pool.ntp.org

2. Установить необходимые пакеты

Команды:

```
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor --yes -o /etc/apt/keyrings/docker.gpg
```

3. Добавить в /etc/apt/sources.list.d/docker.list с помощью команд

Команды:

```
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu bionic stable
apt-get update
apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

4. Настроить автоматический запуск сервиса docker

Команда: systemctl enable --now docker

5. Запустить сервис docker

Команда: systemctl start docker

6. Создать пользователя для установки

Команда: useradd gims

7. Предоставить права пользователю на запуск контейнеров

Команда: usermod gims -aG docker

2.1.4 Подготовка к установке на Red Hat

Внимание!!! Шаги подготовки к установке необходимо выполнить на каждом сервере где будут установлены компоненты GIMS.

1. Синхронизировать время сервера

Команда: ntpdate pool.ntp.org

2. Установить необходимые пакеты

Команды: yum install docker-compose-v2 docker-ce

3. Настроить автоматический запуск сервиса docker

Команда: systemctl enable --now docker

4. Запустить сервис docker

Команда: systemctl start docker

5. Создать пользователя для установки

Команда: useradd gims

6. Предоставить права пользователю на запуск контейнеров

Команда: usermod gims -aG docker

2.1.5 Подготовка к установке на CentOS

Внимание!!! Шаги подготовки к установке необходимо выполнить на каждом сервере где будут установлены компоненты GIMS.

1. Синхронизировать время сервера

Команда: ntpdate pool.ntp.org

2. Установить необходимые пакеты

Команды: yum install docker-compose-v2 docker-ce

3. Настроить автоматический запуск сервиса docker

Команда: systemctl enable --now docker

4. Запустить сервис docker

Команда: systemctl start docker

5. Создать пользователя для установки

Команда: useradd gims

6. Предоставить права пользователю на запуск контейнеров

Команда: usermod gims -aG docker

2.1.6 Подготовка к установке на Debian

Внимание!!! Шаги подготовки к установке необходимо выполнить на каждом сервере где будут установлены компоненты GIMS.

1. Синхронизировать время сервера

Команда: ntpdate pool.ntp.org

2. Установить необходимые пакеты

Команда: apt-get install docker-compose-v2 docker-ce

3. Настроить автоматический запуск сервиса docker

Команда: systemctl enable --now docker

4. Запустить сервис docker

Команда: systemctl start docker

5. Создать пользователя для установки

Команда: useradd gims

6. Предоставить права пользователю на запуск контейнеров

Команда: usermod gims -aG docker

2.2 УСТАНОВКА МОДУЛЕЙ GIMS

2.2.1 Загрузка образа GIMS из репозитория Gelarm

ВАЖНО!!!

- Для выполнения ниже перечисленных команд необходимо получить *username* и *password* для доступа в репозиторий *Gelarm*.
- Шаги по загрузке образов необходимо выполнить на каждом сервере, где будут установлены компоненты *GIMS*.
- Локальный пользователь должен состоять в группе *docker*

1. Получить перечень доступных версий.

Команда: `curl https://registry.gelarm.ru/v2/docker/gims-monitoring/tags/list -u 'username'`

Примечание: Также перечень доступных версий можно получить авторизовавшись в <https://nexus.gelarm.ru>

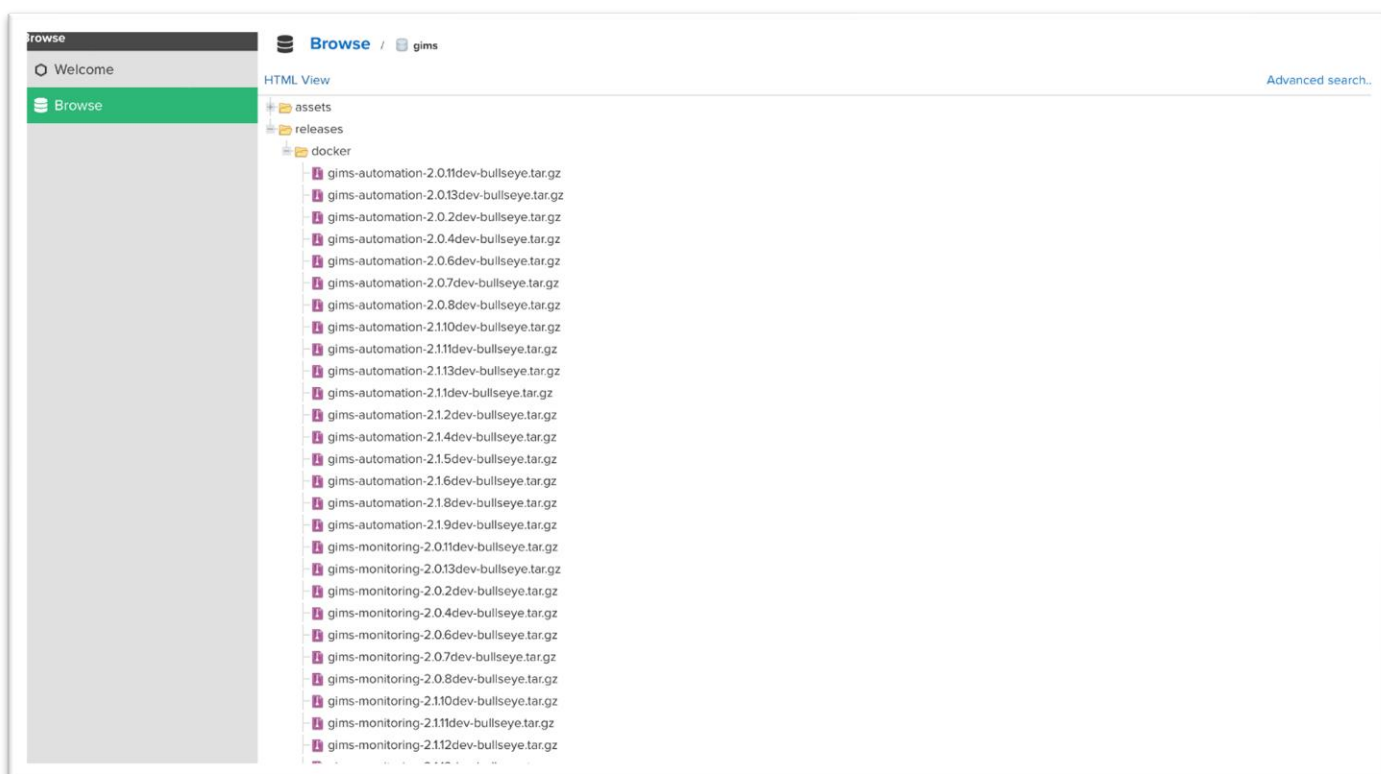


Рисунок 2 – Просмотр доступных версий

2. Загрузить образ GIMS из репозитория.

2.1. Для загрузки образа на локальный сервер использовать следующую команду:

Команда:

`docker login registry.gelarm.ru`

Username: 'username'

Password: 'password'

`docker pull registry.gelarm.ru/docker/'<Название_устанавливаемой_сборки>'`

, где <Название_устанавливаемой_сборки> - это одной из доступных версий полученных на шаге 1

2.2. Для загрузки образа на локальный сервер использовать следующую команду:

Команда:
`docker login registry.gelarm.ru`
`Username: 'username'`
`Password: 'password'`

`docker image save registry.gelarm.ru/docker/'<Название_устанавливаемой_сборки>' | ssh 'root@<адрес_сервера >' docker image load`

`docker image load -i '<Путь_к_сохранённому_образу>'`
, где <Название_устанавливаемой_сборки> - это название одной из доступных версий полученных на Шаге 1
<адрес_сервера> - адрес удаленного сервера на который надо загрузить образ.

2.3. Также возможно скачать образ авторизовавшись в <https://nexus.gelarm.ru> и загрузить образ на удалённый сервер любыми средствами. После загрузки выполнить команду для загрузки его в локальный репозиторий docker.

Команда: `docker image load -i '<Путь_к_сохранённому_образу>'`

3. Создать директорию для установки компонента GIMS

Комада: `mkdir -p /home/gims/gims-docker`

4. Выгрузить файл конфигурации контейнеров из образа

Команда: `docker run --rm --entrypoint=cat docker/gims-monitoring:2.1.35dev-bookworm /home/gims/gims-docker/docker-compose.yaml > docker-compose.yaml`

2.2.2 Установка модуля GIMS Portal

1. Создать файл конфигурации GIMS Portal

Команда:
`cd /home/gims/gims-docker`
`nano portal.sh`

Содержимое файла:
`export ACCEPT_EULA=yes`

```

export GIMS_IMAGE=registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye
export POSTGRES_IMAGE=registry.gelarm.ru/postgres:16
export COMPOSE_PROJECT_NAME=gims-portal
export COMPOSE_PROFILES=portal
export SERVER_IP=10.178.0.131
export SECRET_KEY=8d204b6568e9d990afdf5681676c2a6f
export CLR_LOG_LEVEL=error
exec docker compose "$@"

```

Примечание: Полное описание параметров и их значений приведено в Приложении А.

Таблица 3 – Описание основных параметров файла конфигурации GIMS Portal

ПЕРЕМЕННАЯ	ПРИМЕР ЗНАЧЕНИЯ	ОПИСАНИЕ
ACCEPT_EULA	yes	(required) Согласие с условиями лицензионного соглашения GIMS.
GIMS_IMAGE	registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye	(required) Образ GIMS используемый для установки.
COMPOSE_PROJECT_NAME	gims-portal-1	(required) Имя и порядковый номер устанавливаемого компонента.
COMPOSE_PROFILES	portal	(required) Тип устанавливаемого компонента.
SERVER_IP	10.178.0.131	(optional) IP адрес сервера на который выполняется установка. Требуется в случае одновременной установки нескольких компонент GIMS на один сервер.
SECRET_KEY	8d204b6568e9d990afdf5681676c2a6f	(required) Секретный ключ для взаимодействия компонентов GIMS. Команда: echo \$RANDOM md5sum head -c 32; echo

2. Установить компоненты GIMS Portal

Команда:

```
cd /home/gims/gims-docker
bash portal.sh up -d --wait
```

3. Проверить статус компонентов GIMS Portal

Команда:

```
cd /home/gims/gims-docker
bash portal.sh ps
```

Примечание: при корректной установке все контейнеры должны быть в статусе *healthy*

4. Запустить компоненты модуля GIMS Portal

Команда:

```
cd /home/gims/gims-docker
bash portal.sh start '<Имя_сервиса>'

, где <Имя_сервиса> из Шага
```

5. Доступ к web интерфейсу GIMS Portal

Команда:

```
https://'адрес сервера'
Пользователь: admin
Пароль: 123QWErty
```

2.2.3 Установка модуля GIMS Automation

1. Создать файл конфигурации GIMS Automation

Команда:

```
cd /home/gims/gims-docker
nano automation.sh
```

Содержимое файла:

```
export ACCEPT_EULA=yes
export GIMS_IMAGE=registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye
export POSTGRES_IMAGE=registry.gelarm.ru/postgres:16
export COMPOSE_PROJECT_NAME=gims-automation-node-1
export COMPOSE_PROFILES=automation
export CLUSTER_TYPE=autm
export CLUSTER_ID=2
```

```

export SERVER_ID=2
export SERVER_IP=10.178.0.132
export SECRET_KEY=8d204b6568e9d990afdf5681676c2a6f
export CLR_LOG_LEVEL=error
export CLR_WORKER_MAX_CNT_ACTIVATOR=20
export CLR_WORKER_MAX_CNT_SCRIPT=20
export CLR_ACTIVATOR_MIN_TCP_PORT=4000
export CLR_ACTIVATOR_MAX_TCP_PORT=4100
export RABBITMQ_HOSTNAME=as-gims1-02
export RABBITMQ_CLUSTER_HOSTNAME=rabbit@as-gims1-01
export CLUSTER_HOST1=as-gims1-02
export CLUSTER_HOST2=as-gims1-03
export CLUSTER_HOST3=as-gims1-04
export CLUSTER_HOST4=as-gims2-01
export CLUSTER_HOST5=as-gims2-02
export CLUSTER_HOST1_IP=10.178.0.132
export CLUSTER_HOST2_IP=10.178.0.133
export CLUSTER_HOST3_IP=10.178.0.134
export CLUSTER_HOST4_IP=10.178.0.141
export CLUSTER_HOST5_IP=10.178.0.142
exec docker compose "$@"

```

Примечание: Полное описание параметров и их значений приведено в Приложении А.

Таблица 4 – Описание основных параметров файла конфигурации GIMS Automation

ПЕРЕМЕННАЯ	ПРИМЕР ЗНАЧЕНИЯ	ОПИСАНИЕ
ACCEPT_EULA	yes	(required) Согласие с условиями лицензионного соглашения GIMS
GIMS_IMAGE	registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye	(optional) Образ GIMS используемый для установки.
COMPOSE_PROJECT_NAME	gims-automation-node-1	(required) Имя и порядковый номер устанавливаемого компонента
COMPOSE_PROFILES	automation	(required) Тип устанавливаемого компонента
CLUSTER_TYPE	autm	(required) Тип кластера. Для automation - autm.
SERVER_IP	10.178.0.132	(optional) IP адрес сервера на который выполняется установка. Требуется в случае одновременной установки нескольких компонент GIMS на один сервер

ПЕРЕМЕННАЯ	ПРИМЕР ЗНАЧЕНИЯ	ОПИСАНИЕ
SECRET_KEY	8d204b6568e9d990afdf5681676c2a6f	(required) Секретный ключ для взаимодействия компонентов GIMS. Ключ необходимо использовать, который был создан в разделе “Установка GIMS Portal”
CLUSTER_ID	2	(required) Идентификатор кластера. Настраивается в разделе “Настройка инфраструктуры компонентов GIMS”
SERVER_ID	2	(required) Идентификатор сервера. Настраивается в разделе “Настройка инфраструктуры компонентов GIMS”
CLR_WORKER_MAX_CNT_ACTIVATOR	20	Число рабочих процессов для запуска активаторов
CLR_WORKER_MAX_CNT_SCRIPT	20	Число рабочих процессов для запуска сценариев автоматизации
CLR_ACTIVATOR_MIN_TCP_PORT	4000	Нижняя граница диапазона пробрасываемых портов в контейнер активаторов
CLR_ACTIVATOR_MAX_TCP_PORT	4100	Верхняя граница диапазона пробрасываемых портов в контейнер активаторов
RABBITMQ_HOSTNAME	as-gims1-02	(required) Короткое имя текущего узла в кластере Automation. Должно разрешаться через /etc/hosts или ДНС.
RABBITMQ_CLUSTER_HOSTNAME	rabbit@as-gims1-01	(required для всех узлов кластера, кроме первого) Короткое имя первого узла в кластере Automation. Должно разрешаться через /etc/hosts или ДНС.
Формат: rabbit@		
CLUSTER_HOST1, CLUSTER_HOST2, CLUSTER_HOST3 ...	as-gims1-02	Имена всех узлов кластера Automation, которые будут введены в кластер. параметр указывается, если имена не разрешаются через ДНС.
CLUSTER_HOST1_IP, CLUSTER_HOST2_IP, CLUSTER_HOST3_IP ...	10.178.0.132	IP всех узлов кластера Automation, которые будут введены в кластер. параметр указывается, если имена не разрешаются через ДНС.

2. Установить компонент модуля GIMS Automation

Команда:

```
cd /home/gims/gims-docker
bash automation.sh up -d --wait
```

3. Проверить статус компонентов модуля GIMS Automation

Команда:

```
cd /home/gims/gims-docker
bash automation.sh ps
```

Примечание: при корректной установке все контейнеры должны быть в статусе healthy

4. Запуск компонентов модуля GIMS Automation

Команда:

```
cd /home/gims/gims-docker
bash automation.sh start '<Имя_сервиса>'
```

2.2.4 Установка модуля GIMS Inventory

1. Создать файл конфигурации GIMS Inventory

Команда:

```
cd /home/gims/gims-docker
nano inventory.sh
```

Содержимое файла:

```
export ACCEPT_EULA=yes
export GIMS_IMAGE=registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye
export POSTGRES_IMAGE=registry.gelarm.ru/postgres:16
export COMPOSE_PROJECT_NAME=gims-inventory-node-1
export COMPOSE_PROFILES=inventory
export CLUSTER_TYPE=invdb
export CLUSTER_ID=3
export SERVER_ID=9
export SERVER_IP=10.178.0.143
export SECRET_KEY=8d204b6568e9d990afdf5681676c2a6f
exec docker compose "$@"
```

Примечание: Полное описание параметров и их значений приведено в Приложении А.

Таблица 5 – Описание основных параметров файла конфигурации GIMS Inventory

ПЕРЕМЕННАЯ	ПРИМЕР ЗНАЧЕНИЯ	ОПИСАНИЕ
ACCEPT_EULA	yes	(required) Согласие с условиями лицензионного соглашения GIMS
GIMS_IMAGE	registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye	(optional) Образ GIMS используемый для установки.
COMPOSE_PROJECT_NAME	gims-inventory-node-1	(required) Имя и порядковый номер устанавливаемого компонента
COMPOSE_PROFILES	inventory	(required) Тип устанавливаемого компонента
CLUSTER_TYPE	invdb	(required) Тип кластера. Для inventory - invdb.
SERVER_IP	10.178.0.143	(optional) IP адрес сервера на который выполняется установка. Требуется в случае одновременной установки нескольких компонент GIMS на один сервер
SECRET_KEY	8d204b6568e9d990afdf5681676c2a6f	(required) Секретный ключ для взаимодействия компонентов GIMS. Ключ необходимо использовать, который был создан в разделе “Установка GIMS Portal”
CLUSTER_ID	3	(required) Идентификатор кластера. Настраивается в разделе “Настройка инфраструктуры компонентов GIMS”
SERVER_ID	9	(required) Идентификатор сервера. Настраивается в разделе “Настройка инфраструктуры компонентов GIMS”

2. Установить компоненты модуля GIMS Inventory

Команда:
`cd /home/gims/gims-docker`
`bash inventory.sh up -d --wait`

3. Проверить статус компонентов модуля GIMS Inventory

Команда:
`cd /home/gims/gims-docker`
`bash inventory.sh ps`

Примечание: при корректной установке все контейнеры должны быть в статусе healthy

4. Запустить компоненты модуля GIMS Inventory

Команда:
`cd /home/gims/gims-docker`
`bash inventory.sh start 'имя сервиса'`

2.2.5 Установка модуля GIMS Monitoring Fault

1. Создать файл конфигурации GIMS Monitoring Fault

Команда:
`cd /home/gims/gims-docker`
`nano faultdb.sh`

Содержимое файла:
`export ACCEPT_EULA=yes`
`export GIMS_IMAGE=registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye`
`export POSTGRES_IMAGE=registry.gelarm.ru/postgres:16-cron`
`export COMPOSE_PROJECT_NAME=gims-faultdb-node-1`
`export COMPOSE_PROFILES=faultdb`
`export CLUSTER_TYPE=fltdb`
`export CLUSTER_ID=3`
`export SERVER_ID=9`
`export SERVER_IP=10.178.0.144`
`export SECRET_KEY=8d204b6568e9d990afdf5681676c2a6f`
`exec docker compose "$@"`

Примечание: Полное описание параметров и их значений приведено в Приложении А.

Таблица 6 – Описание основных параметров файла конфигурации GIMS Monitoring Fault

ПЕРЕМЕННАЯ	ПРИМЕР ЗНАЧЕНИЯ	ОПИСАНИЕ
ACCEPT_EULA	yes	(required) Согласие с условиями лицензионного соглашения GIMS
GIMS_IMAGE	registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye	(optional) Образ GIMS используемый для установки.
COMPOSE_PROJECT_NAME	gims-faultdb-node-1	(required) Имя и порядковый номер устанавливаемого компонента

ПЕРЕМЕННАЯ	ПРИМЕР ЗНАЧЕНИЯ	ОПИСАНИЕ
COMPOSE_PROFILES	faultdb	(required) Тип устанавливаемого компонента
CLUSTER_TYPE	fltdb	(required) Тип кластера. Для faultdb - fltdb.
SERVER_IP	10.178.0.144	(optional) IP адрес сервера на который выполняется установка. Требуется в случае одновременной установки нескольких компонент GIMS на один сервер
SECRET_KEY	8d204b6568e9d990afdf5681676c2a6f	(required) Секретный ключ для взаимодействия компонентов GIMS. Ключ необходимо использовать, который был создан в разделе “Установка GIMS Portal”
CLUSTER_ID	3	(required) Идентификатор кластера. Настраивается в разделе “Настройка инфраструктуры компонентов GIMS”
SERVER_ID	9	(required) Идентификатор сервера. Настраивается в разделе “Настройка инфраструктуры компонентов GIMS”

2. Установить компоненты модуля Monitoring Fault

Команда:
`cd /home/gims/gims-docker`
`bash faultdb.sh up -d –wait`

3. Проверить статус компонентов модуля GIMS Monitoring Fault

Команда:
`cd /home/gims/gims-docker`
`bash faultdb.sh ps`

Примечание: при корректной установке все контейнеры должны быть в статусе healthy

4. Запустить компоненты модуля GIMS Monitoring Fault

Команда:
`cd /home/gims/gims-docker`
`bash faultdb.sh start 'имя сервиса'`

2.2.6 Установка модуля GIMS Monitoring Performance

1. Создать файл конфигурации GIMS Monitoring Performance

Команда:

```
cd /home/gims/gims-docker  
nano perfmon.sh
```

Содержимое файла:

```
export ACCEPT_EULA=yes  
export GIMS_IMAGE=registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye  
export POSTGRES_IMAGE=registry.gelarm.ru/postgres:16-timescaledb-2  
export COMPOSE_PROJECT_NAME=gims-perfmon-node-1  
export COMPOSE_PROFILES=perfmon  
export CLUSTER_TYPE=prfmn  
export CLUSTER_ID=3  
export SERVER_ID=9  
export SERVER_IP=10.178.0.145  
export INVENTORY_SERVER=192.168.108.59  
export SECRET_KEY=8d204b6568e9d990afdf5681676c2a6f  
exec docker compose "$@"
```

Примечание: Полное описание параметров и их значений приведено в Приложении А.

Таблица 7 – Описание основных параметров файла конфигурации GIMS Monitoring Performance

ПЕРЕМЕННАЯ	ПРИМЕР ЗНАЧЕНИЯ	ОПИСАНИЕ
ACCEPT_EULA	yes	(required) Согласие с условиями лицензионного соглашения GIMS
GIMS_IMAGE	registry.gelarm.ru/docker/gims-monitoring:2.0.13dev-bullseye	(optional) Образ GIMS используемый для установки.
POSTGRES_IMAGE	registry.gelarm.ru/timescale/timescaledb:2.11.2-pg15	(required) Образ БД с возможностью хранения временных рядов
COMPOSE_PROJECT_NAME	gims-perfmon-node-1	(required) Имя и порядковый номер устанавливаемого компонента
COMPOSE_PROFILES	faultdb	(required) Тип устанавливаемого компонента
CLUSTER_TYPE	fltdb	(required) Тип кластера. Для perfmon - prfmn.
SERVER_IP	10.178.0.144	(optional) IP адрес сервера на который выполняется установка. Требуется в

ПЕРЕМЕННАЯ	ПРИМЕР ЗНАЧЕНИЯ	ОПИСАНИЕ
		случае одновременной установки нескольких компонент GIMS на один сервер
INVENTORY_SERVER	192.168.108.59	(required) IP адрес сервера GIMS Inventory
SECRET_KEY	8d204b6568e9d990afdf5681676c2a6f	(required) Секретный ключ для взаимодействия компонентов GIMS. Ключ необходимо использовать, который был создан в разделе “Установка GIMS Portal”
CLUSTER_ID	3	(required) Идентификатор кластера. Настраивается в разделе “Настройка инфраструктуры компонентов GIMS”
SERVER_ID	9	(required) Идентификатор сервера. Настраивается в разделе “Настройка инфраструктуры компонентов GIMS”

2. Установить компоненты модуля Monitoring Performance

Команда:
`cd /home/gims/gims-docker`
`bash perfmon.sh up -d –wait`

3. Проверить статус компонентов модуля Monitoring Performance

Команда:
`cd /home/gims/gims-docker`
`bash perfmon.sh ps`

Примечание: при корректной установке все контейнеры должны быть в статусе healthy

4. Запустить компоненты модуля GIMS Monitoring Performance

Команда:
`cd /home/gims/gims-docker`
`bash perfmon.sh start 'имя сервиса'`

2.3 НАСТРОЙКА ИНФРАСТРУКТУРЫ КОМПОНЕНТОВ GIMS

1. Создать кластер GIMS Portal одним из следующих способов:

1.1. Выполнить команду на сервере, на котором установлен компонент GIMS Portal

Команда: `bash portal.sh exec -it backend gimsctl cluster_add --type '<Tun_параметра>' --connection 1 --name "Portal"`
, где <Tun_параметра> - параметр из таблицы ниже

ЗНАЧЕНИЕ ПАРАМЕТРОВ TYPE	ОПИСАНИЕ
1	Кластер GIMS Portal. В инфраструктуре возможно наличие только одного кластера.
2	Кластер GIMS Automation. В инфраструктуре возможно наличие неограниченного количества кластеров.
3	Кластер GIMS Inventory. В инфраструктуре возможно наличие только одного кластера.
4	Кластер GIMS Fault. В инфраструктуре возможно наличие неограниченного количества кластеров.
5	Кластер GIMS Performance. В инфраструктуре возможно наличие только одного кластера.

1.2. Также создать кластер возможно на странице “Конфигуратор инфраструктуры” GIMS Portal.

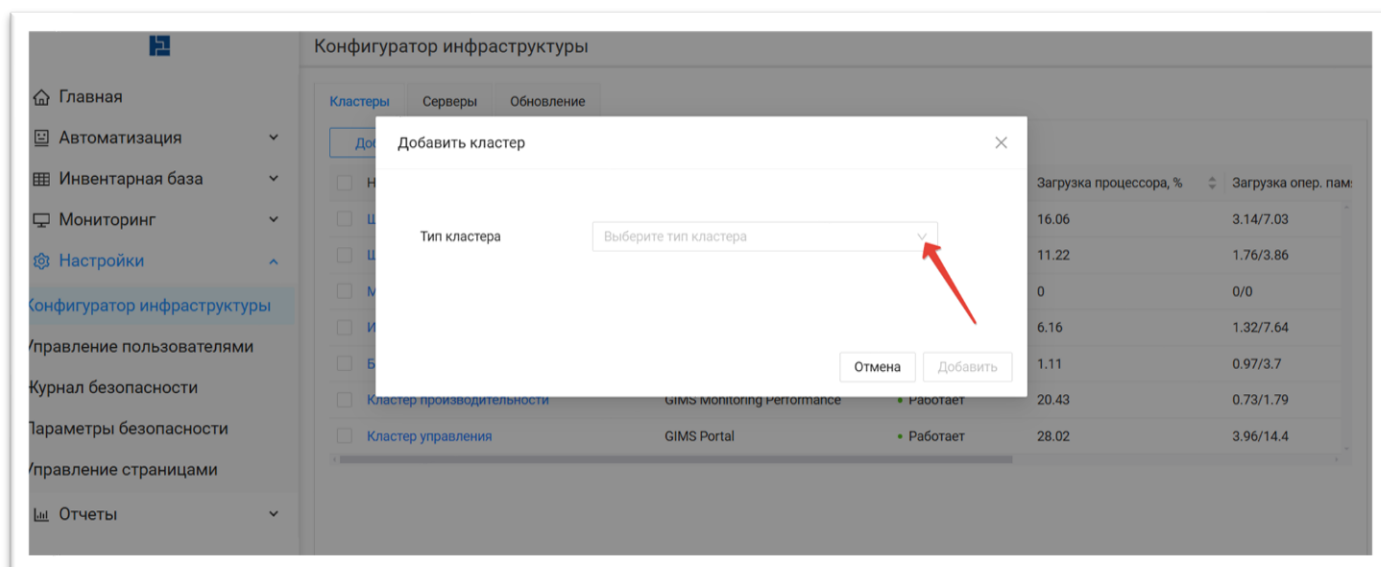


Рисунок 3 – Добавление кластера GIMS Portal через интерфейс

2. Добавление сервера в кластер

2.1. Выполнить команду на сервере, на котором установлен компонент GIMS Portal

Команда: `bash portal.sh exec -it backend gimsctl server_add --cluster '<Идентификатор_кластера>' --address '<ip_адрес>' --name "<Имя>"`

, где

<Идентификатор_кластера> - Задаёт идентификатор кластера в который необходимо добавить сервер

<ip_адрес> - Задаёт ip адрес или DNS имя сервера

<Имя> - Задаёт имя сервера для отображения в списке серверов конфигуратора

2.2. Также добавить сервер в кластер возможно на странице “Конфигуратор инфраструктуры” GIMS Portal.

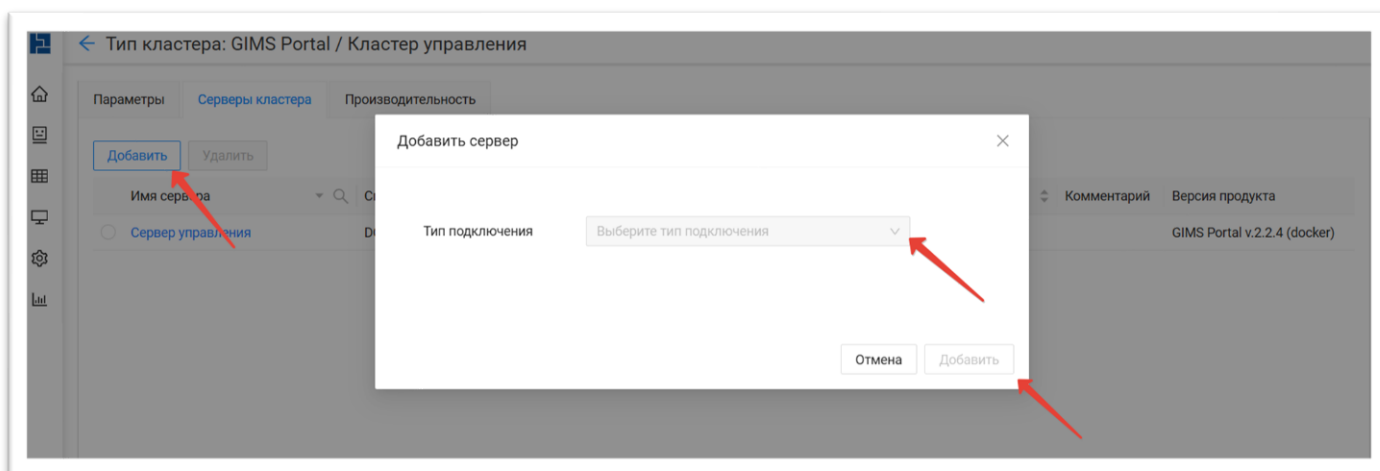


Рисунок 4 – Добавление Сервера GIMS Portal через интерфейс

2.4 КОМАНДЫ ДЛЯ УПРАВЛЕНИЯ УСТАНОВЛЕННЫМИ МОДУЛЯМИ

Таблица 8 – Список дополнительных команд для управления Системой

НАЗНАЧЕНИЕ КОМАНДЫ	ТЕКСТ КОМАНДЫ
Справка по команде управления инфраструктурой	<code>bash portal.sh exec -it backend gimsctl --help</code>
Вывести список серверов (Данная команда выполняется на сервере, на котором установлен компонент GIMS Portal)	<code>\$ bash portal.sh exec backend gimsctl server_list</code>
Остановка компонентов модуля GIMS Portal	<code>cd /home/gims/gims-docker</code> <code>bash portal.sh stop '<Имя_сервиса>'</code> , где <Имя_сервиса> из Шага
Удаление компонентов GIMS Portal	<code>cd /home/gims/gims-docker</code> <code>bash portal.sh down -v -t 0</code>

НАЗНАЧЕНИЕ КОМАНДЫ	ТЕКСТ КОМАНДЫ
Остановка компонентов модуля GIMS Automation	cd /home/gims/gims-docker bash automation.sh stop '<Имя_сервиса>' , где <Имя_сервиса> из Шага
Удаление компонентов GIMS Automation	cd /home/gims/gims-docker bash automation.sh down -v -t 0
Остановка компонентов GIMS Inventory	cd /home/gims/gims-docker bash inventory.sh stop '<Имя_сервиса>' , где <Имя_сервиса> из Шага
Удаление компонентов GIMS Inventory	cd /home/gims/gims-docker bash inventory.sh down -v -t 0
Остановка компонентов GIMS Monitoring Fault	cd /home/gims/gims-docker bash faultdb.sh stop '<Имя_сервиса>' , где <Имя_сервиса> из Шага
Удаление компонентов GIMS Monitoring Fault	cd /home/gims/gims-docker bash faultdb.sh down -v -t 0
Остановка компонентов GIMS Monitoring Performance	cd /home/gims/gims-docker bash perfmon.sh stop '<Имя_сервиса>' , где <Имя_сервиса> из Шага
Удаление компонентов GIMS Monitoring Performance	cd /home/gims/gims-docker bash perfmon.sh down -v -t 0

Приложение А. ПАРАМЕТРЫ НАСТРОЙКИ КОМПОНЕНТ

НАЗВАНИЕ ПАРАМЕТРА	НАЗВАНИЕ КОМПОНЕНТА	НАЗВАНИЕ КОНТЕЙНЕРА	ОПИСАНИЕ	ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ
SYSLOG_LOGROTATE_COUNT	Portal, Automation,	syslog	Указывает сколько старых логов нужно хранить.	10
SYSLOG_LOGROTATE_SIZE	Portal, Automation.	syslog	Размер лога, когда он будет перемещен в мегабайтах.	100M
SYSLOG_SYSLOGD_LOGFORMAT	Portal, Automation.	syslog	Формат лога.	%syslogtag%%msg%\n
PERFMON_APPLY_LOGLEVEL	Performance	perfmon-apply	Уровень логирования в контейнере.	INFO
PERFMON_APPLY_PERIOD	Performance	perfmon-apply	Время синхронизации и применения порогов в секундах.	60
CLR_ACTIVATOR_MIN_TCP_PORT	Automation		Нижняя граница диапазона пробрасываемых портов в контейнер активаторов	4000
CLR_ACTIVATOR_MAX_TCP_PORT	Automation		Верхняя граница диапазона пробрасываемых портов в контейнер активаторов	4100
CLR_SCHEDULER_WORKER_MAX_CNT	Automation		Максимально количество воркеров расписания.	10
CLR_SCHEDULER_LOGGER_LEVEL	Automation		Уровень логирования воркера расписания.	'INFO'
CLR_SCHEDULER_JOB_COALESCE	Automation			'True'

НАЗВАНИЕ ПАРАМЕТРА	НАЗВАНИЕ КОМПОНЕНТА	НАЗВАНИЕ КОНТЕЙНЕРА	ОПИСАНИЕ	ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ
CLR_SCHEDULER_JOB_MAX_INSTANCES	Automation		Максимальное количество одновременно запущенных задач для каждого расписания.	1
CLR_SCHEDULER_JOB_MISFIRE_GRACE_TIME	Automation			60
AGENT_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-1000
POSTGRES_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-1000
PGBOUNCER_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-1000
RABBITMQ_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-1000
FRONTEND_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-700
BACKEND_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-700
BACKEND_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-700
CELERY_SCRIPT_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-500
CELERY_CRON_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-500
CELERY_DEPLOY_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-500
CELERY_BEAT_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-500

НАЗВАНИЕ ПАРАМЕТРА	НАЗВАНИЕ КОМПОНЕНТА	НАЗВАНИЕ КОНТЕЙНЕРА	ОПИСАНИЕ	ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ
CELERY_ACTIVATOR_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-500
CELERY_PREAMACTIVATOR_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-500
CELERY_SCHEDULER_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-500
PERFMON_INIT_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-700
PERFMON_EVENTS_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-700
PERFMON_ALERTS_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-700
PERFMON_APPLY_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-700
REPLICATOR_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-700
MONITOR_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-700
UPDATE_OOM_SCORE_ADJ			Приоритет процесса при работе OOM killer	-1000
FRONTEND_SSL_CRT	Portal	Frontend	Файл ssl сертификата портала	\$(cat /path/to/cert.crt base64 -w0)
FRONTEND_SSL_KEY	Portal	Frontend	Файл ключа ssl сертификата портала	\$(cat /path/to/cert.key base64 -w0)

Название параметра	Название компонента	Название контейнера	Описание	Значение по умолчанию
KRB5_CONFIG_DATA	Portal		Файл настроек krb5.config	\$(cat /path/to/krb5.config base64 -w0)
KRB5_KEYTAB_DATA	Portal		Файл key tab	\$(cat /path/to/krb5.keytab base64 -w0)