**ПРОГРАММНЫЙ ПРОДУКТ**

**«NEXT GENERATION SECURE CLOUD (NGSC): ЗАЩИЩЕННОЕ ЧАСТНОЕ ОБЛАКО»**

**Руководство по развертыванию**

2024

**Аннотация**

Данный документ предназначен для специалистов, выполняющих администрирование программного продукта «Next Generation Secure Cloud (NGSC): Защищенное Частное Облако» (далее – **Программный продукт**), и включает описание действий по установке и настройке Программного продукта в операционных системах, поддерживающих систему управления контейнерами.

Описание функциональных возможностей и компонентов Программного продукта, а также требования к аппаратному и программному обеспечению содержатся в документе «NGSC Описание системы».

Инструкции по администрированию и использованию Программного продукта содержатся в документе «NGSC Руководство по эксплуатации».

Содержание

[1 Перед запуском Программного продукта 4](#_Toc161767801)

[1.1 Подготовка дистрибутива 4](#_Toc161767802)

[1.2 Подготовка файла переменных 4](#_Toc161767803)

[1.3 Создание сертификатов 10](#_Toc161767804)

[1.4 Запуск Private Docker Registry с дистрибутивом решения 10](#_Toc161767805)

[1.5 Первоначальная настройка платформы виртуализации РУСТЭК 11](#_Toc161767806)

[2 Запуск решения в Kubernetes 13](#_Toc161767807)

[2.1 Проверка готовности компонентов 13](#_Toc161767808)

[2.2 Настройка компонентов 13](#_Toc161767809)

[2.3 Запуск решения 16](#_Toc161767810)

[3 Создание пользователей 17](#_Toc161767811)

[Перечень используемых сокращений 18](#_Toc161767812)

# Перед запуском Программного продукта

## Подготовка дистрибутива

Перед установкой Программного продукта выполните подготовку дистрибутива.

Скачайте дистрибутив на сайте производителя. Распакуйте архив ngsc\_v.1.0.tar.

Убедитесь в том, что:

1. появилась папка Deploy со скриптами установки и что она имеет именно такое название;
2. появилась папка release с образами Docker:
   1. bitnami\_kafka-3.6.1.tar.gz
   2. caddy-2-alpine.tar.gz
   3. elasticsearch-8.11.1.tar.gz
   4. kibana-8.11.1.tar.gz
   5. netboxcommunity\_netbox-v3.6.tar.gz
   6. ngsc\_conductor\_server-3.15.0.tar.gz
   7. ngsc\_conductor\_ui-3.15.0.tar.gz
   8. ngsc\_dashboard\_api-1.0.2.tar.gz
   9. ngsc\_dashboard\_app-1.0.2.tar.gz
   10. ngsc\_init-services-1.0.tar.gz
   11. ngsc\_keycloak-22.0.tar.gz
   12. ngsc\_openbao-1.14.9.tar.gz
   13. ngsc\_postgres-15.tar.gz
   14. ngsc\_sambadc-4.tar.gz
   15. redis-7-alpine.tar.gz

## Подготовка файла переменных

Подготовьте файл переменных. Возьмите за основу следующий пример и подставьте свои переменные. Обратите внимание на то, что в файле примера проставлены значения по умолчанию и для примера. Убедитесь в том, что указанные значения, например, IP-адреса локальных серверов, доменное имя, не конфликтуют с вашей инфраструктурой. В случае конфликта укажите в файле переменных значения, которые вы планируете использовать. Укажите пароли, которые вы планируете использовать. Переменные, которые должны быть обязательно изменены, выделены. Сохраните готовый файл переменных в Deploy\environments.env.**$Environment**.

|  |
| --- |
| # Общие настройки  **PFX\_PASSWORD**="" # Пароль для файлов pkcs12 (форматы pfx, p12)  **CA\_PFX\_PASSWORD**="" # Пароль для файлов pkcs12 корневого сертификата (Root CA)  SOLUTION\_NAME="NGSC" # Название вашего решения (используется в описаниях Root CA)  REGISTRY\_HOST="registry" # Имя хоста для Docker registry  **REGISTRY\_USER**=" " # Имя пользователя для доступа к Docker registry  **REGISTRY\_USER\_PASSWORD**="" # Пароль для доступа к Docker registry  # Настройки для Kubernetes  NAMESPACE="ngsc" # Пространство имен в Kubernetes  **NFS\_SERVER**="127.0.0.1" # IPv4 адрес вашего NFS сервера  **NFS\_SERVER\_DATA\_PATH**="/data/ngsc" # Путь на NFS сервере, где хранятся постоянные данные  # Настройки для правильного разрешения имени до сервера виртуализации  **RUSTACK\_HOST**="rustack" # Имя сервера виртуализации РУСТЭК  **RUSTACK\_HOST\_IP**="" # IPv4 адрес сервера виртуализации РУСТЭК  # Настройки Samba-DC  DOMAIN="inncloud" # NETBIOS Доменное имя  SAMBA\_HOST="pdc" # Имя хоста Samba  DOMAIN\_GROUP\_PREFIX="inn" # Префикс групп для сопоставления с Keycloak. Т.е. будут созданы группы innUsers, innGroups  DOMAIN\_FQDN="inncloud.local" # Полное доменное имя (FQDN). FQDN применяется для всех хостов, используемых Программным продуктом  **DOMAIN\_ADMIN\_PASSWORD**="" # Пароль администратора домена  DOMAIN\_DN\_PATH="DC=inncloud,DC=local" # Distinguished Name (DN) домена. Он создается из полного доменного имени (FQDN) -переменная DOMAIN\_FQDN, путем разбиения его по точке и добавления префикса "DC=" к каждой части. Например, для FQDN "inncloud.local", DOMAIN\_DN\_PATH будет "DC=inncloud,DC=local". Это должно строго соответствовать FQDN.  DOMAIN\_GLOBAL\_ADMIN="glob\_admin" # Глобальный администратор домена. **Не менять**  **DOMAIN\_GLOBAL\_ADMIN\_PASSWORD**="" # Пароль глобального администратора  SHOULD\_USE\_SAMBA\_HOST="${SAMBA\_HOST}" # Используется для корректной работы подстановки переменных. **Не менять**  SHOULD\_USE\_DOMAIN\_ADMIN\_PASSWORD="${DOMAIN\_ADMIN\_PASSWORD}" # Используется для корректной работы подстановки переменных. **Не менять**  # Настройка базы данных PostgreSQL  POSTGRES\_HOST="postgres" # Имя хоста PostgreSQL  POSTGRES\_EXT\_NODEPORT="30541" # NodePort для внешнего подключения к PostgreSQL  POSTGRES\_PORT="5432" # Порт для подключения к PostgreSQL. **Не менять**  **POSTGRES\_ADMIN**=" " # Имя администратора PostgreSQL  **POSTGRES\_ADMIN\_PASSWORD**="" # Пароль администратора PostgreSQL  # Настройки Keycloak  KEYCLOAK\_HOST="keycloak" # Имя хоста Keycloak  KEYCLOAK\_DB\_NAME="keycloak" # Имя базы данных Keycloak  KEYCLOAK\_DB\_USER="keycloak" # Имя пользователя для подключения к базе данных Keycloak  **KEYCLOAK\_DB\_PASSWORD**="" # Пароль пользователя для подключения к базе данных Keycloak  REALM\_NAME="NGSC" # Имя области (Realm) в Keycloak. Удобнее, если REALM\_NAME будет совпадать со значением **SOLUTION\_NAME**  SOLUTION\_NAME\_LOW="ngsc" # Название вашего решения в нижнем регистре (используется как префикс для клиентов) – **должно строго соответствовать значению SOLUTION\_NAME**  KEYCLOAK\_ADMIN="admin" # Имя встроенного администратора Keycloak  **KEYCLOAK\_ADMIN\_PASSWORD**="" # Пароль встроенного администратора Keycloak  DOMAIN\_KEYCLOAK\_SVC\_CONNECTOR="svc-keycloak" # Сервисный коннектор Keycloak для домена  **DOMAIN\_KEYCLOAK\_SVC\_CONNECTOR\_PASSWORD**="" # Пароль сервисного коннектора Keycloak для домена  # Настройка Elasticsearch и Kibana (для Api Dashboard)  ELASTICSEARCH\_HOST="elasticsearch" # Имя хоста Elasticsearch  KIBANA\_HOST="kibana" # Имя хоста Kibana  KIBANA\_SYSTEM\_USER="ngsckibana\_system" # Имя сервисной учетной записи для подключения Kibana к **Elasticsearch**  **KIBANA\_SYSTEM\_PASSWORD**="" # Пароль сервисной учетной записи для подключения Kibana к Elasticsearch  KIBANA\_ADMIN\_USER="ngsckibana" # Имя пользователя для работы с Kibana. Используйте это имя для входа на Kibana  **KIBANA\_ADMIN\_USER\_PASSWORD**="" # Пароль пользователя для работы с Kibana  # Настройки API Dashboard  API\_DASHBOARD\_HOST="api" # Имя хоста для API Dashboard  API\_DASHBOARD\_DB\_NAME="ngsc" # Имя базы данных для Dashboard  API\_DASHBOARD\_DB\_USER="dashboard" # Имя пользователя для доступа к базе данных Dashboard  **API\_DASHBOARD\_DB\_PASSWORD**="" # Пароль пользователя для доступа к базе данных Dashboard  API\_DASHBOARD\_PORT="443" # Порт для API Dashboard. **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_Cluster\_\_MinValue="0" # Диапазон адресов «от». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_Cluster\_\_MaxValue="199" # Диапазон адресов «до». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_Tenant\_\_MinValue="0" # Диапазон адресов «от». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_Tenant\_\_MaxValue="199" # Диапазон адресов «до». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_BusinessGroup\_\_MinValue="0" # Диапазон адресов «от». **Не менять** API\_DASHBOARD\_CodeValueRange\_\_BusinessGroup\_\_MaxValue="199"# Диапазон адресов «до». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_InformationSystem\_\_MinValue="0" # Диапазон адресов «от». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_InformationSystem\_\_MaxValue="199" # Диапазон адресов «до». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_VirtualNetwork\_\_MinValue="0" # Диапазон адресов «от». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_VirtualNetwork\_\_MaxValue="199" # Диапазон адресов «до». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_VirtualMachine\_\_MinValue="0" # Диапазон адресов «от». **Не менять** API\_DASHBOARD\_CodeValueRange\_\_VirtualMachine\_\_MaxValue="199" # Диапазон адресов «до». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_NetworkRule\_\_MinValue="0"# Диапазон адресов «от». **Не менять**  API\_DASHBOARD\_CodeValueRange\_\_NetworkRule\_\_MaxValue="199" # Диапазон адресов «до». **Не менять**  API\_DASHBOARD\_VERSION="1.0.0" # Версия API Dashboard. Именно эта версия image будет использоваться для сервиса: ${REGISTRY}.${DOMAIN\_FQDN}/ngsc/dashboard/api:${API\_DASHBOARD\_VERSION}  DASHBOARD\_ELASTIC\_USER="apidashboarduser" # Имя пользователя для подключения Serilog к Elasticsearch  **DASHBOARD\_ELASTIC\_USER\_PASSWORD**="" # Пароль пользователя для подключения Serilog к Elasticsearch  DASHBOARD\_PORT="443" # Порт для Dashboard. **Не менять**  # Настройки Dashboard UI  DASHBOARD\_HOST="dashboard" # Имя хоста для Dashboard  DASHBOARD\_PORT="443" # Порт для Dashboard. **Не менять**  DASHBOARD\_USER="dashboard-user" # Имя доменного пользователя для работы с Dashboard. Можно использовать этого пользователя для входа в сервис  **DASHBOARD\_PASSWORD**="" # Пароль доменного пользователя для работы с Dashboard  DASHBOARD\_VERSION="1.0.0" # Версия Dashboard. Именно эта версия image будет использоваться для сервиса: ${REGISTRY}.${DOMAIN\_FQDN}/ngsc/dashboard/app:${DASHBOARD\_VERSION}  # Настройки Redis  REDIS\_HOST="redis" # Имя хоста для Redis с AOF (Append Only File) - метод, который использует Redis для постоянных данных.  REDIS\_PORT="6379" # Порт для Redis. **Не менять**  REDIS\_SSL\_PORT="7379" # Порт для Redis с TLS. **Не менять**  **REDIS\_PASSWORD**="" # Пароль для подключения к Redis  REDIS\_CACHE\_HOST="redis-cache" # Имя хоста для Redis Cache с RDB  REDIS\_CACHE\_PORT="6379" # Порт для Redis Cache  **REDIS\_CACHE\_PASSWORD**="" # Пароль для подключения к Redis Cache  SHOULD\_USE\_REDIS\_PASSWORD="$REDIS\_PASSWORD" # Используется для корректной работы подстановки переменных. **Не менять**  # Настройки Netbox  NETBOX\_HOST="netbox" # Имя хоста для Netbox  NETBOX\_PORT="8080" # Порт для Netbox. **Не менять**  **NETBOX\_PASSWORD**="" # Пароль администратора Netbox  NETBOX\_DB\_NAME="netbox" # Имя базы данных Netbox  NETBOX\_DB\_USER="netbox" # Имя пользователя для доступа к базе данных Netbox  **NETBOX\_DB\_PASSWORD**="" # Пароль пользователя для доступа к базе данных Netbox  **NETBOX\_SECRET\_KEY**="" # Секретный ключ Netbox. Секретный ключ используется для обеспечения криптографической безопасности различных функций в Netbox, таких как подпись cookies  **NETBOX\_SUPERUSER\_PASSWORD**="" # Пароль встроенного суперпользователя Netbox  DOMAIN\_NETBOX\_SVC\_CONNECTOR="svc-netbox" # Имя сервисного коннектора Netbox для домена  **DOMAIN\_NETBOX\_SVC\_CONNECTOR\_PASSWORD**="" # Пароль сервисного коннектора Netbox для домена  DOMAIN\_NETBOX\_USER="netbox-user" # Имя доменной учетной записи обычного пользователя Netbox  **DOMAIN\_NETBOX\_USER\_PASSWORD**="" # Пароль доменной учетной записи обычного пользователя Netbox  DOMAIN\_NETBOX\_ADMIN\_USER="netbox-user-admin" # Имя доменной учетной записи административного пользователя Netbox  **DOMAIN\_NETBOX\_ADMIN\_USER\_PASSWORD**="" # Пароль доменной учетной записи административного пользователя Netbox  # Настройка Kafka  KAFKA\_HOST="kafka" # Имя хоста Kafka  KAFKA\_PORT="9092" # Порт для подключения к Kafka  KAFKA\_JVM\_PERFORMANCE\_OPTS="-XX:MetaspaceSize=96m -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercent=35 -XX:G1HeapRegionSize=16M -XX:MinMetaspaceFreeRatio=50 -XX:MaxMetaspaceFreeRatio=80" # Опции JVM для оптимизации производительности Kafka. Не рекомендуется менять.  KAFKA\_HEAP\_OPTS="-Xmx512m -Xms512m" # Опции Heap для Kafka Не рекомендуется менять  KAFKA\_OPTS="-server -XX:+UseG1GC -XX:MaxGCPauseMillis=20" # Дополнительные опции для Kafka. Не рекомендуется менять  **KAFKA\_KRAFT\_CLUSTER\_ID**="" # ID кластера Kafka Kraft  KAFKA\_CFG\_OFFSETS\_TOPIC\_REPLICATION\_FACTOR="3" # Фактор репликации для темы смещений Kafka  KAFKA\_CFG\_TRANSACTION\_STATE\_LOG\_REPLICATION\_FACTOR="3" # Фактор репликации для журнала состояния транзакций Kafka  KAFKA\_CFG\_TRANSACTION\_STATE\_LOG\_MIN\_ISR="2" # Минимальное значение ISR для журнала состояния транзакций Kafka  KAFKA\_CLIENT\_USERS="user" # Имя пользователя для подключения клиентов к Kafka  **KAFKA\_CLIENT\_PASSWORDS**="" # Пароль пользователя для подключения клиентов к Kafka  KAFKA\_INTER\_BROKER\_USER="inter\_broker\_user" # Имя пользователя для межброкерного взаимодействия в Kafka  **KAFKA\_INTER\_BROKER\_PASSWORD**="" # Пароль пользователя для межброкерного взаимодействия в Kafka  KAFKA\_CONTROLLER\_USER="controller\_user" # Имя пользователя для контроллера Kafka  KAFKA\_CONTROLLER\_PASSWORD="" # Пароль пользователя для контроллера Kafka  KAFKA\_KRAFT\_BOOTSTRAP\_SCRAM\_USERS="True" # Использовать ли SCRAM пользователей для бутстрапа Kafka Kraft  # Настройка Conductor  WF\_SERVER\_HOST="wf-server" # Имя хоста для сервера рабочего процесса Conductor  WF\_SERVER\_DB\_NAME="conductor" # Имя базы данных для сервера рабочего процесса Conductor  WF\_SERVER\_DB\_USER="conductor" # Имя пользователя для подключения к базе данных сервера рабочего процесса Conductor  **WF\_SERVER\_DB\_PASSWORD**="" # Пароль пользователя для подключения к базе данных сервера рабочего процесса Conductor  JAVA\_TOOL\_OPTIONS="-Xms512m -Xmx1536m -XX:+UseG1GC -XX:+UseStringDeduplication" # Опции Java для оптимизации производительности  CONDUCTOR\_UI\_HOST="conductor" # Имя хоста для пользовательского интерфейса Conductor  CONDUCTOR\_UI\_PORT="5000" # Порт для пользовательского интерфейса Conductor. **Не менять**  # Настройка Vault  VAULT\_HOST="vault" # Имя хоста для Vault  VAULT\_PORT="8200" # Порт для подключения к Vault  VAULT\_PREFIX="inno" # Префикс для хранилища секретов Vault  DOMAIN\_VAULT\_SVC\_CONNECTOR="svc-vault" # Сервисный коннектор Vault для домена  DOMAIN\_VAULT\_SVC\_CONNECTOR\_PASSWORD="" # Пароль сервисного коннектора Vault для домена  DOMAIN\_VAULT\_ADMIN\_USER="vault-admin" # Имя административного пользователя Vault для домена  **DOMAIN\_VAULT\_ADMIN\_PASSWORD**="" # Пароль административного пользователя Vault для домена  VAULT\_EXT\_NODEPORT="30821" # Node Port для внешнего подключения к Vault |

После создания файла подгрузите его в переменные окружения консоли:

**bash:**

|  |
| --- |
| source Deploy/scripts/init\_variables.sh $Environment |

**powershell:**

|  |
| --- |
| Deploy\scripts\init\_variables.ps1 $Environment |

## Создание сертификатов

Подгрузите переменные с помощью init\_variables.

Создайте папку**:**

|  |
| --- |
| mkdir certs |

Создайте запросы на сертификаты CSR для подписи на Enterprise PKI:

**bash:**

|  |
| --- |
| Deploy/scripts/generate\_certificates\_PKI.sh |

**powershell:**

|  |
| --- |
| Deploy\scripts\generate\_certificates\_PKI.ps1 |

Результатом выполнения скриптов будет ряд запросов (файлы csr+key) на сертификат с закрытым ключом RSA, которые необходимо подписать на вашем Enterprise PKI (сервер инфраструктуры открытых ключей).

Сохраните все .key и .crt файлы в формате кодирования сертификатов x.509 (PEM) (и в папке с расширением .crt и .key:

|  |
| --- |
| public certificate: ./certs/${HOSTNAME}.${DOMAIN\_FQDN}.crt  private cerficate key: ./certs/${HOSTNAME}.${DOMAIN\_FQDN}.key |

## Запуск Private Docker Registry с дистрибутивом решения

Если в окружении для будущего развертывания решения присутствует Private Docker Registry, необходимо подгрузить tar архивы с помощью Docker и выгрузить images в Private Registry. Для выполнения этих действий можно использовать скрипт «load and push», описание которого есть в шапке исполняемого скрипта:

|  |
| --- |
| Deploy/scripts/load-push.sh |

Если в окружении для будущего развертывания решения отсутствует Private Docker Registry, то его необходимо развернуть. Для этого существует два варианта:

1. Отдельная машина с Private Docker Registry

Выполните скрипт для разворачивания Private Docker Registry:

|  |
| --- |
| Deploy/scripts/setup-dedicated\_registry.sh |

В результате выполнения скрипта создастся машина с Private Docker Registry.

1. Ресурс в Kubernetes с Private Docker Registry

Выполните скрипт для разворачивания Private Docker Registry в Kubernetes:

**bash:**

|  |
| --- |
| Deploy/kubernetes/scripts/setup-registry.sh |

**powershell:**

|  |
| --- |
| Deploy\kubernetes\scripts\setup-registry.ps1 |

*Примечание: В качестве Private Docker Registry используется образ Registry:2.*

Перед стартом убедитесь в том, что:

* Подготовлены сертификаты для Private Docker Registry;
* Private Docker Registry доступен для целевой платформы по протоколу https.

Доступ к Private Docker Registry осуществляется по учетным данным в файле окружения:

$REGISTRY\_USER;

$REGISTRY\_USER\_PASSWORD.

## Первоначальная настройка платформы виртуализации РУСТЭК

Внесите изменения в настройки по умолчанию платформы РУСТЭК. Эти изменения необходимы для корректной работы Программного продукта.

**Настройка сервиса NEURON:**

* Измените количество маршрутов на роутерах. Для этого на всех узлах с ролью "Управление сетями" в секцию [DEFAULT] в файле /etc/neutron/neutron.conf укажите следующее значение для параметра:

max\_routes = 200

Перезапустите сервис neutron-server.

* Измените количество сетевых квот. Для этого на всех узлах с ролью "Управление сетями" в секцию [quotas] в файл /etc/neutron/neutron.conf укажите следующие значения для параметров:

quota\_security\_group = 50

quota\_security\_group\_rule = 500

Перезапустите сервис neutron-server.

**Настройка сервиса NOVA:**

Измените максимальное количество дисков виртуальной машины. Для этого на всех узлах с ролью "Вычислительный узел" в секции [compute] в файле /etc/nova/nova.conf укажите следующее значение для параметра:

max\_disk\_devices\_to\_attach = 21

Перезапустите сервис nova-compute.

**Создание служебных пользователей:**

Создайте служебного пользователя платформы РУСТЭК со следующими свойствами:

Имя: ngsc-sa

Права: admin в платформе, admin в домене Default, admin в домене Default с наследованием, admin в проекте admin

Пароль: любой

**Настройка агрегатов:**

Удостоверьтесь в том, что на платформе создан как минимум 1 агрегат.

**Шаблоны виртуальных машин:**

В качестве средств создания шаблонов виртуальных машин поддерживается РЕД ОС 7.3. Убедитесь в том, что служба cloud-init активирована в образе, а также что образ с этой ОС имеет следующие свойства и метаданные:

Имя: 'RedOS 7.3.2'

Свойство 'image\_type' = 'ngsc'

Свойство 'distro' = 'RedOS 7.3'

Свойство 'public' = true

# Запуск решения в Kubernetes

## Проверка готовности компонентов

Убедитесь в том, что у вас присутствуют следующие подготовленные компоненты:

* NFS Storage Server развернут в сети;
* Подготовлен файл переменных;
* Подготовлены сертификаты;
* Подготовлен хост с kubectl;
* Развернут и доступен по сети Private Docker Registry.

## Настройка компонентов

1. Kubectl

Скрипты должны запускаться с машины, на которой настроен kubectl, есть доступ к API кластера.

Для настройки удаленного доступа kubectl необходимо настроить kubeconfig файл:

Для Windows: C:\Users\$($env:USERNAME)\.kube\config;

Для Linux: /user/.kube/config.

1. NFS Storage Server

Проверьте настройки NFSv4 сервера:

|  |
| --- |
| cat /etc/export |

|  |
| --- |
| /data 10.0.0.4/25(rw,insecure,nohide,no\_root\_squash,no\_subtree\_check)  /data 10.0.0.5/25(rw,insecure,nohide,no\_root\_squash,no\_subtree\_check)  /data 10.0.0.6/25(rw,insecure,nohide,no\_root\_squash,no\_subtree\_check) |

В папке $NFS\_SERVER\_DATA\_PATH (согласно файлу переменных) создайте необходимые для создания persistent volume папки:

|  |
| --- |
| sudo mkdir elasticsearch  sudo chmod 777 elasticsearch  sudo mkdir kafka  sudo chmod 777 kafka  sudo mkdir kibana  sudo chmod 777 kibana  sudo mkdir postgres  sudo chmod 777 postgres  sudo mkdir redis  sudo chmod 777 redis  sudo mkdir registry  sudo chmod 777 registry  sudo mkdir vault  sudo chmod 777 vault |

1. Доверие к Private Docker Registry

На всех узлах кластера создайте папку и скопируйте rootca.crt:

|  |
| --- |
| sudo mkdir -p /etc/containers/certs.d/$REGISTRY.$DOMAIN\_FQDN  sudo cp Root\_CA.crt /etc/containers/certs.d/$REGISTRY.$DOMAIN\_FQDN/сa.crt на всех нодах |

1. NGINX Ingress Controller

Развернуть NGINX Ingress Controller, если ранее это не было выполнено, согласно документации: https://docs.nginx.com/nginx-ingress-controller/installation/installing-nic/installation-with-manifests/

1. Настройка HAproxy и NGINX Ingress Controller на работу с High Availability IP адресом

Допишите в конфигурацию HAproxy:

|  |
| --- |
| #---------------------------------------------------------------------  # main frontend which proxys to the backends  #---------------------------------------------------------------------  frontend ngsc  mode http  bind \*:80  default\_backend k8s-http  frontend ngsc-ssl  mode tcp  bind \*:443  default\_backend k8s-https  #---------------------------------------------------------------------  # round robin balancing between the various backends  #---------------------------------------------------------------------  backend k8s-http  mode http  balance roundrobin  server k8s-w1 10.0.0.4:30080 check  server k8s-w2 10.0.0.5:30080 check  server k8s-w3 10.0.0.7:30080 check  http-request set-header X-Forwarded-For %[src]  http-request set-header X-Forwarded-Port %[dst\_port]  backend k8s-https  mode tcp  balance roundrobin  server k8s-w1 10.0.0.4:30443 check  server k8s-w2 10.0.0.5:30443 check  server k8s-w3 10.0.0.7:30443 check  http-request set-header X-Forwarded-For %[src]  http-request set-header X-Forwarded-Port %[dst\_port] |

Где: k8s-w{x} – имя worker ноды и 10.{a}.{b}.{c} – IPv4 адрес worker ноды.

Перезапустите сервис на мастер узлах с HAproxy.

Примените новую конфигурацию для NGINX Ingress controller:

|  |
| --- |
| kubectl apply -f Deploy/kubernetes/ingress-controller.yaml |

## Запуск решения

Распакуйте архив с пакетом запуска deploy-ngsc.tar, если этого не было сделано ранее:

|  |
| --- |
| tar -xf deploy-ngsc.tar |

Перейдите в рабочую папку:

|  |
| --- |
| cd ngsc |

Запустите применение Kubernetes манифестов:

**bash:**

|  |
| --- |
| Deploy/kubernetes/scripts/deploy.sh |

**powershell:**

|  |
| --- |
| Deploy\kubernetes\scripts\deploy.ps1 |

Все операции настройки выполняются в автоматическом режиме.

# Создание пользователей

Для создания пользователя выполните следующие действия:

1. Зайдите под учетной записью администратора на <https://keycloak.ngsc.local/>;
2. Перейдите в «Realm»;
3. Перейдите в раздел «Users»;
4. Нажмите кнопку «Add user»;
5. Заполните форму по примеру:

Пример

|  |
| --- |
| Required user actions: пусто  Username: Ngsc.User  Email: Ngsc.User@domain.com  Email verified: Yes  First name: NGSC  Last name: USER |

1. Нажмите кнопку «Create»;
2. Перейдите на вкладку «Credentials»;
3. Нажмите кнопку «Set password»;
4. Введите любой пароль, подтвердите его и установите флаг «Temporary».

Сообщите пользователю ссылку для входа <https://dashboard.ngsc.local/>, его Username и пароль, который необходимо сменить при первом входе.

# Перечень используемых сокращений

| **Сокращение** | **Полное наименование** |
| --- | --- |
| **API** | Application Programming Interface |
| **ОС** | Операционная система |
| **ПО** | Программное обеспечение |
| **Программный продукт** | Программный продукт «Next Generation Secure Cloud (NGSC): Защищенное Частное Облако» |
| **БД** | База данных |
| **СУБД** | Система управления базами данных |