РУСТЭК-ЕСУ

РУКОВОДСТВО
ПО УСТАНОВКЕ И НАСТРОЙКЕ РУСТЭК-ЕСУ

Версия 3.4.2

2022

Оглавление

1. Поставка РУСТЭК-ЕСУ

РУСТЭК-ЕСУ поставляется в виде образа виртуальной машины ESU-box. В зависимости от целевой платформы виртуализации, на которой будет производиться инсталляция, используются форматы:

- .raw – для установки на РУСТЭК (KVM).
- .ova – для установки на VMware ESXi.

В качестве гостевой ОС используется Debian 10 (может меняться производителем). В ESU-box встроен инсталлятор, а также запущены необходимые для работы сервисы и программное обеспечение в виде docker-контейнеров. Это удобно для быстрого запуска Системы.

Минимальные требования для сервера ESU-box:

- vCPU 4 ядра.
- RAM 4 ГБ.
- Размер диска 20 ГБ.

2. Развёртывание на платформе виртуализации РУСТЭК

2.1. Системные требования

- РУСТЭК Wallaby.
- Одна маршрутизируемая сеть, минимально допустимая /27, с доступом до management-сети РУСТЭК

Пример схемы сетевой связности РУСТЭК-ЕСУ, установленной внутри платформы виртуализации РУСТЭК с подключенной к ней инсталляцией VMware vSphere (***Рисунок 1***).

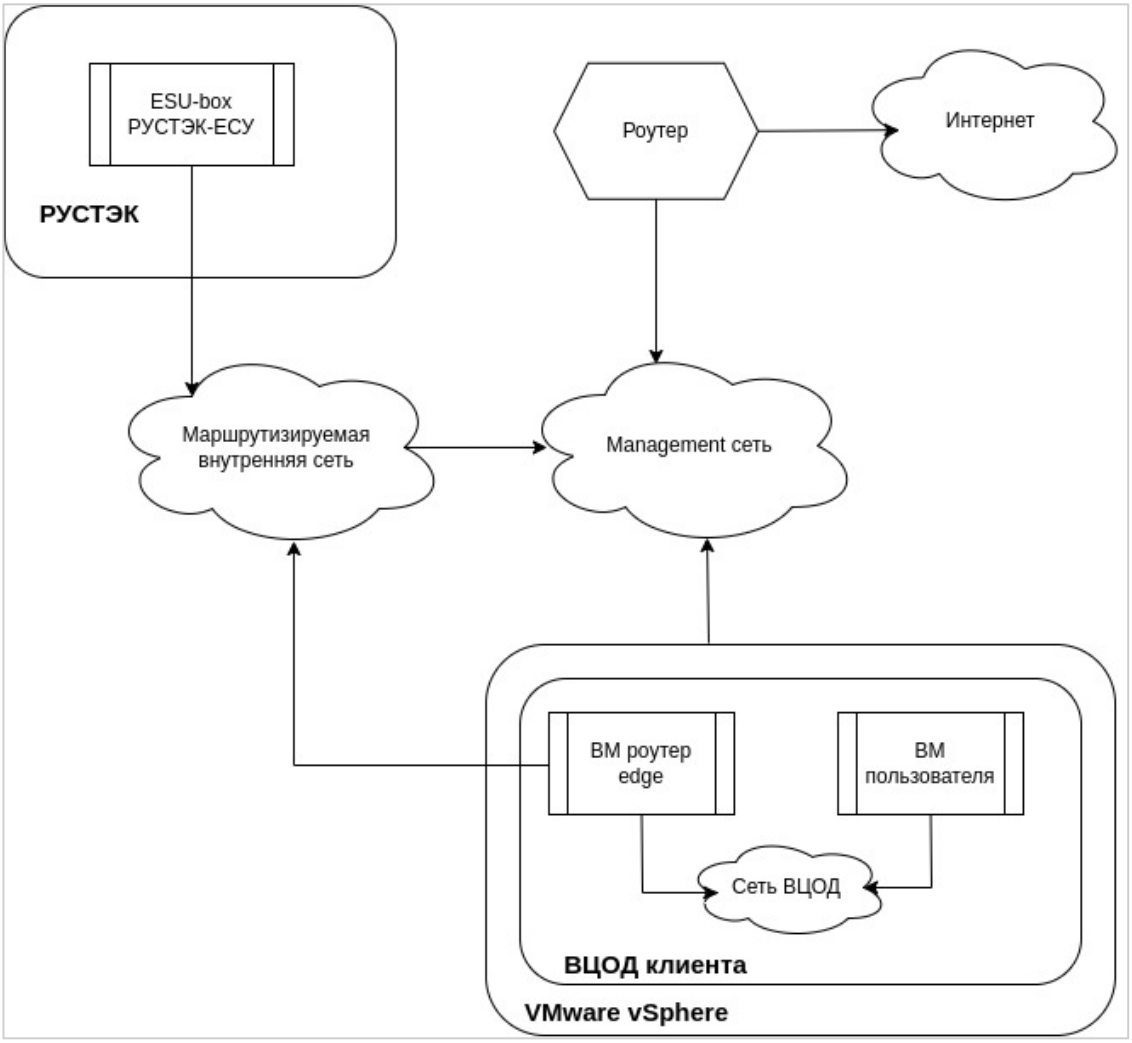

Рисунок 1

2.2. Порядок развертывания

1. Зайти в Панель управления РУСТЭК по ссылке https://<virtual ip>/New
2. Создание образа (***Рисунок 2, Рисунок 3***):

   Перейти в раздел «Копии и образы» – «Образы» и нажать кнопку «Создать».

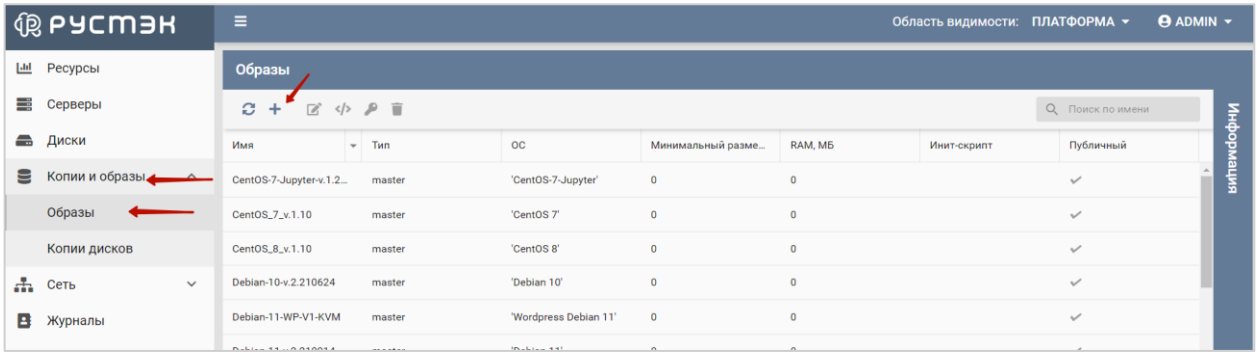

Рисунок 2

Откроется окно с параметрами образа, где необходимо заполнить поля:

- Имя – указывается «произвольное».
- Имя ОС – указывается «произвольное».
- Контейнер – оставить значение «bare».
- Формат диска – указать «raw».
- RAM(МБ) – указывается минимальное кол-во ОЗУ для будущих ВМ – указать 4096.
- Размер диска(ГБ) – указывается минимальный размер дисков для будущих ВМ – указать 20 ГБ.
- Сетевой адаптер – выбрать «virtio».
- Дисковый контроллер – выбрать «virtio-scsi».
- Публичный – снять чек-бокс.
- Метод загрузки – выбрать «Файл».

И нажать «Создать».

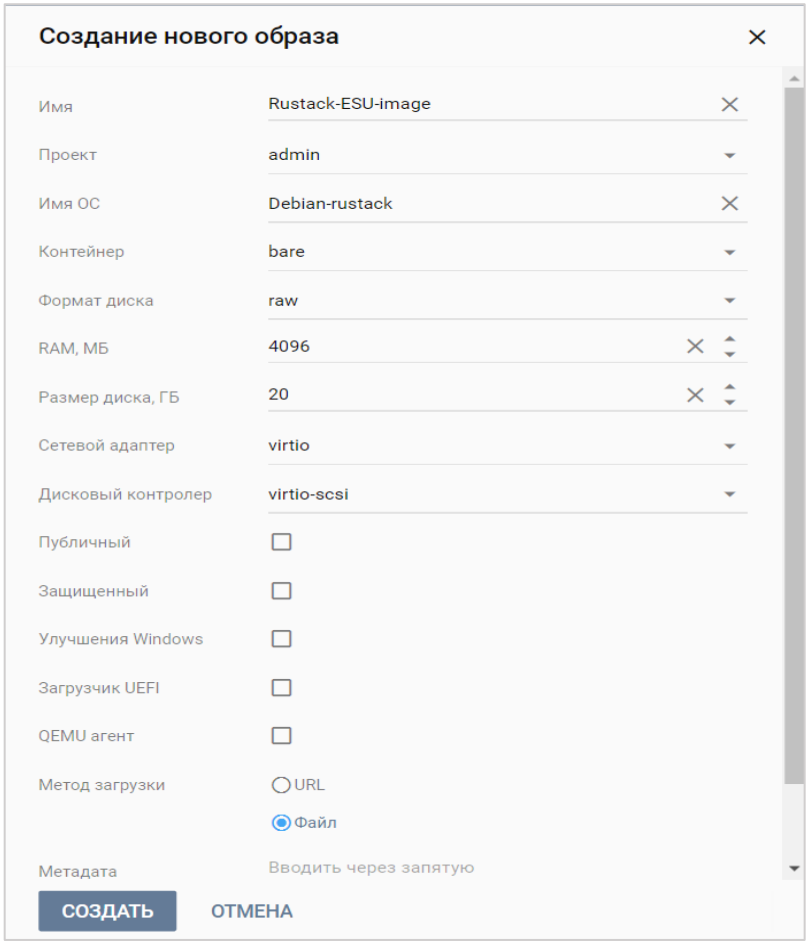

Рисунок 3

3. Загрузка образа (***Рисунок 4, Рисунок 5***)

   Найти в списке новый образ, выбрать его и нажать на кнопку «Загрузить образ».

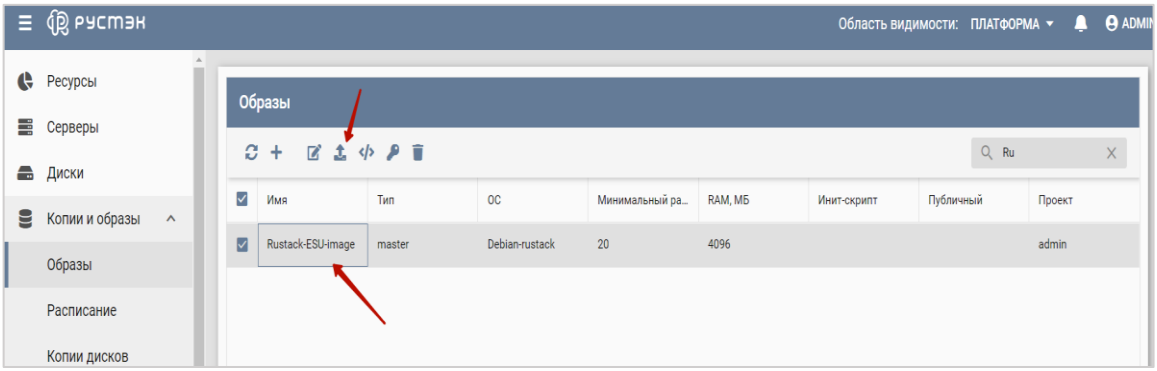

Рисунок 4

Нажать «Добавить файл» и выбрать предоставленный дистрибутив в формате raw. Далее нажать «Загрузить». Начнется процесс загрузки образа.

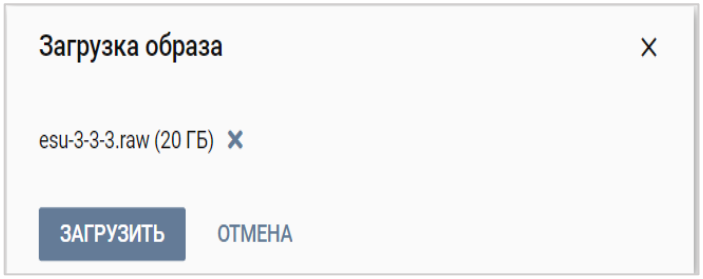

Рисунок 5

4. Создание маршрутизируемой сети (***Рисунок 6***, ***Рисунок 7***)

   По окончании загрузки вам необходимо создать сеть для будущей Единой системы управления. Для этого необходимо перейти в раздел «Сеть» – «Сети» и нажать «Создать».

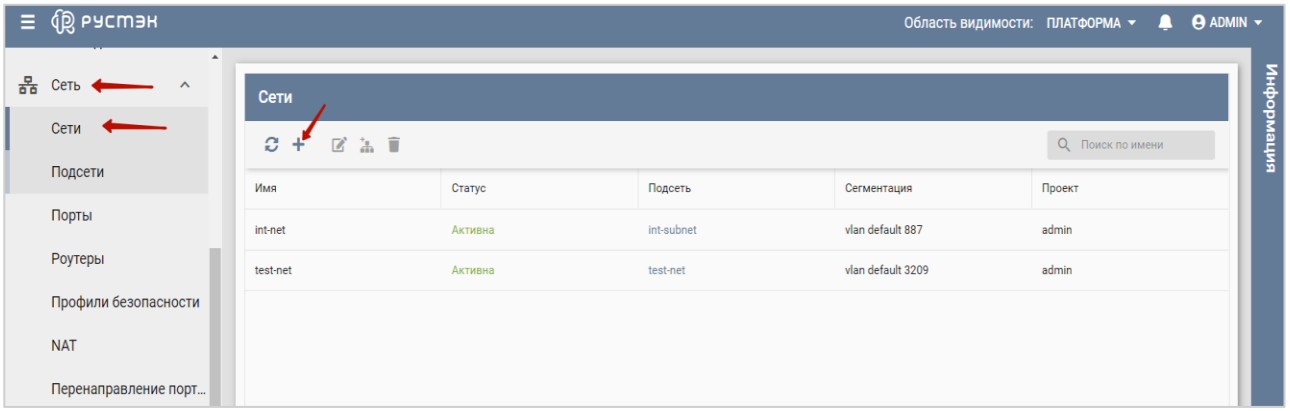

Рисунок 6

Заполнить необходимые поля:

- Имя – указывается произвольное.
- Тип сегментации – VLAN.
- Номер VLAN – номер выделенного влана для менеджмент-сети Единой системы управления.
- Безопасность портов – отключаем.
- Внешняя — ставим галочку.

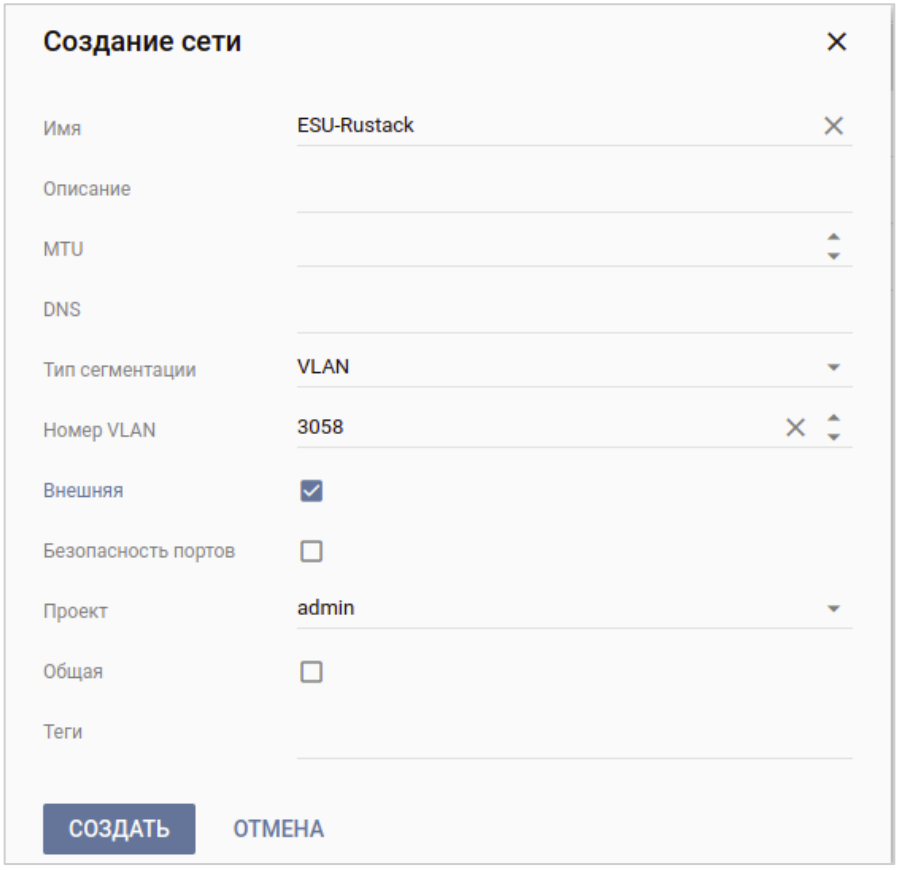

Рисунок 7

5. Создание подсети для маршрутизируемой сети (***Рисунок 8***)

   После создания сети необходимо создать подсеть. Для этого перейдите в раздел «Сети» – «Подсеть» и нажмите «Создать», далее необходимо заполнить поля:

   - Имя – указывается произвольное.
   - Сеть – выбрать сеть, созданную на предыдущем этапе.
   - Версия протокола – Ipv4.
   - Адрес сети – указать cidr.
   - Шлюз – указать шлюз.
   - DHCP – снять чек-бокс.

   Нажать «Создать».

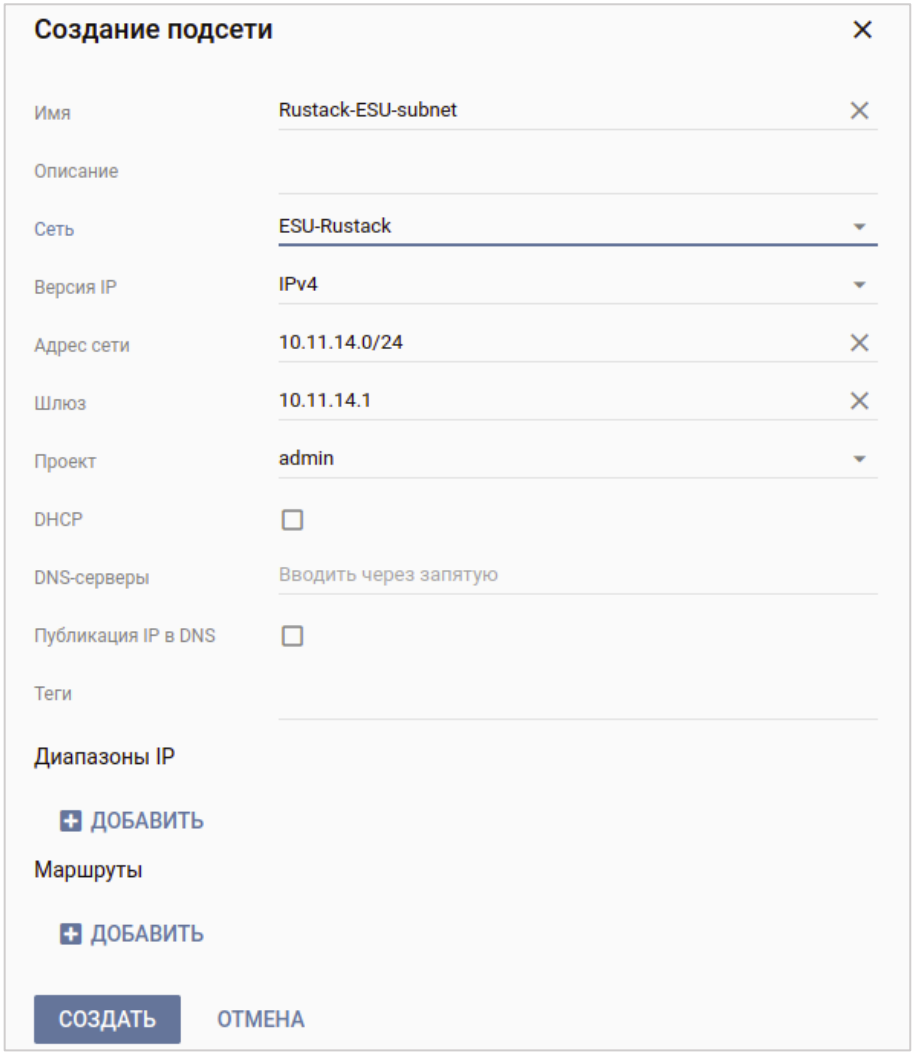

Рисунок 8

!!!Важно!!! Из создаваемой сети для будущей ВМ ESU_box должен быть организован доступ до менеджмент-сети хостов виртуализации.

6. Создание конфигурации ВМ (***Рисунок 9***)

   Перейдите в раздел **Конфигурация** – **Конфигурации** и нажмите «Создать». После чего необходимо заполнить поля будущей конфигурации:

   - Имя – указывается произвольное.
   - vCPU – количество CPU.
   - RAM, МБ – количество ОЗУ. Обратите внимание, что размер указывается в Мб.

   Нажать «Создать».

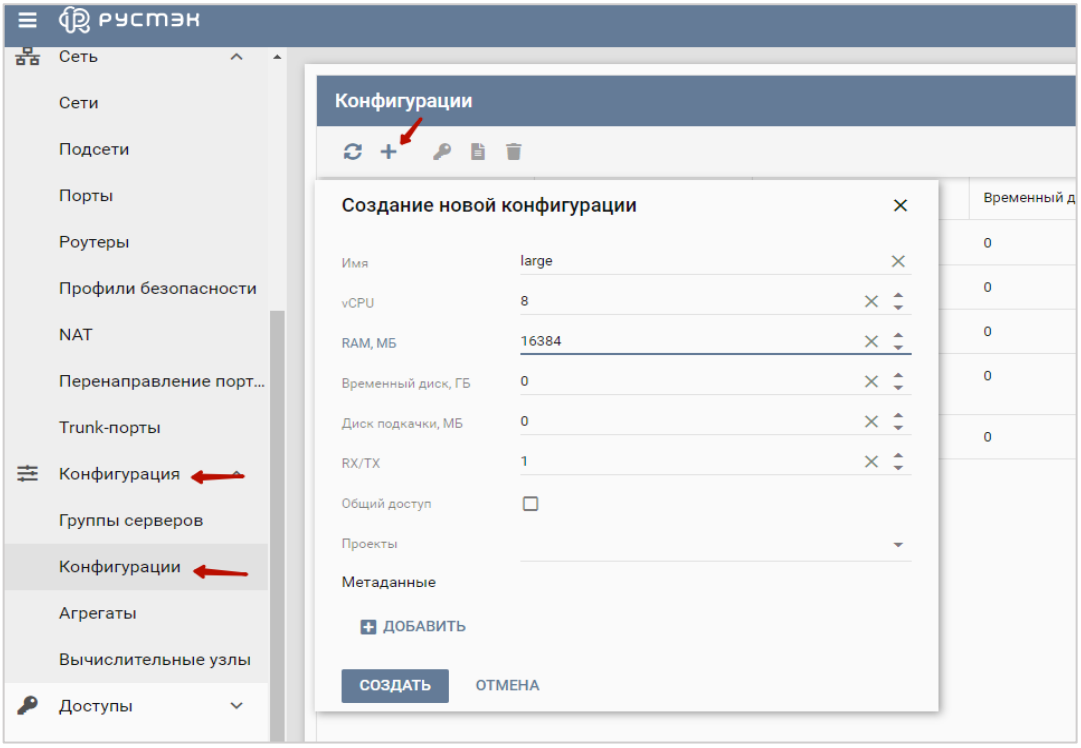

Рисунок 9

7. Создание ВМ (***Рисунок 10***)

Необходимо перейти во вкладку «Серверы» и нажать «Создать» в появившейся форме требуется заполнить поля:

- Имя – указывается произвольное.
- ОС – выбрать ранее загруженный образ.
- Конфигурация – указать необходимую конфигурацию (минимальная 4 CPU, 4ГБ RAM).
- Размер диска – указать размер диска ВМ (минимальный размер 20 Гб).
- Чек-бокс «Удалять диск вместе с сервером» – рекомендуем снять.
- Сети – выбрать ранее созданную маршрутизируемую сеть.

Нажать «Создать».

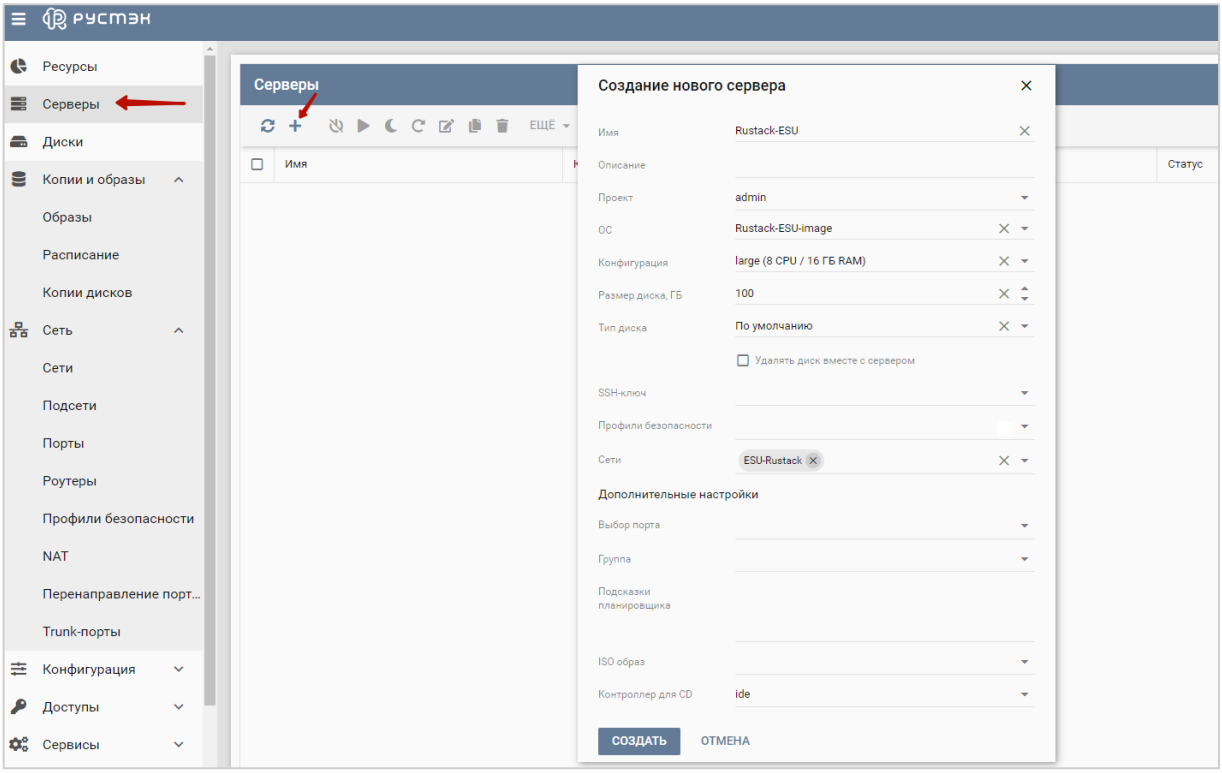

Рисунок 10

Дождаться окончания создания сервера (статус изменится на «Запущен»).

8. Открываем VNC-консоль для созданной ВМ **(*Рисунок 11*)**

   Для открытия консоли сервера переходим в меню «Серверы». Выбираем созданный сервер, затем нажимаем «Ещё» и либо сразу открываем консоль сервера, нажав «Открыть консоль», либо получаем ссылку, нажав на «Ссылка на консоль сервера» и открываем её в новой вкладке.

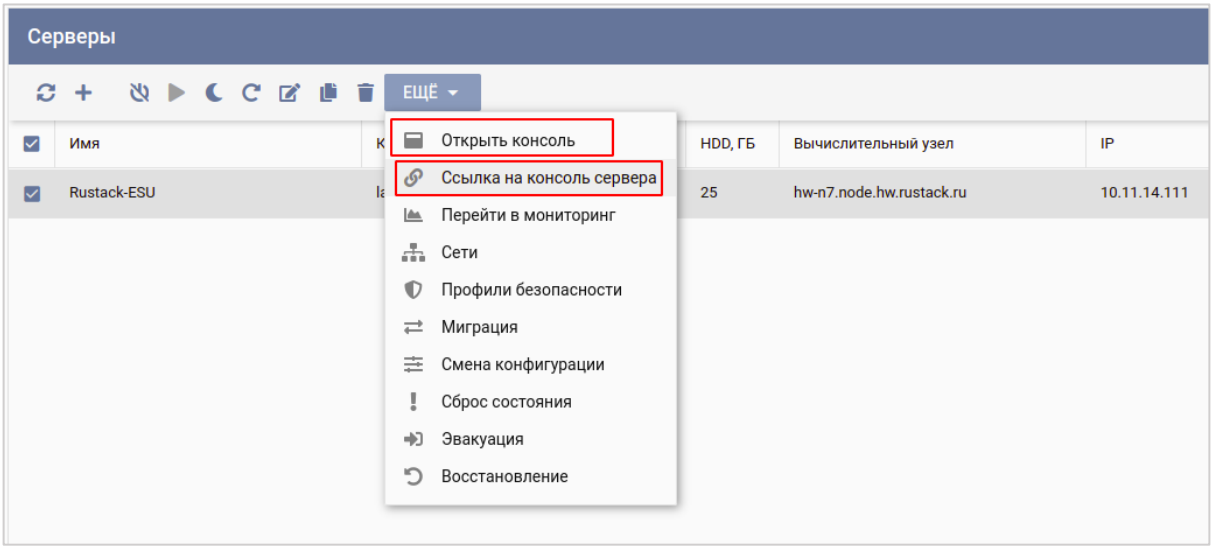

Рисунок 11

Стандартная учётная запись на сервере с РУСТЭК-ЕСУ (ESU-box): **deploy:1-qpALzm/**

3. Установка РУСТЭК-ЕСУ

Установка запускается автоматически при запуске ВМ с РУСТЭК-ЕСУ.

Сначала произойдет распаковка контейнеров. Нужно дождаться завершения процесса (***Рисунок 12*** и ***Рисунок 13***):

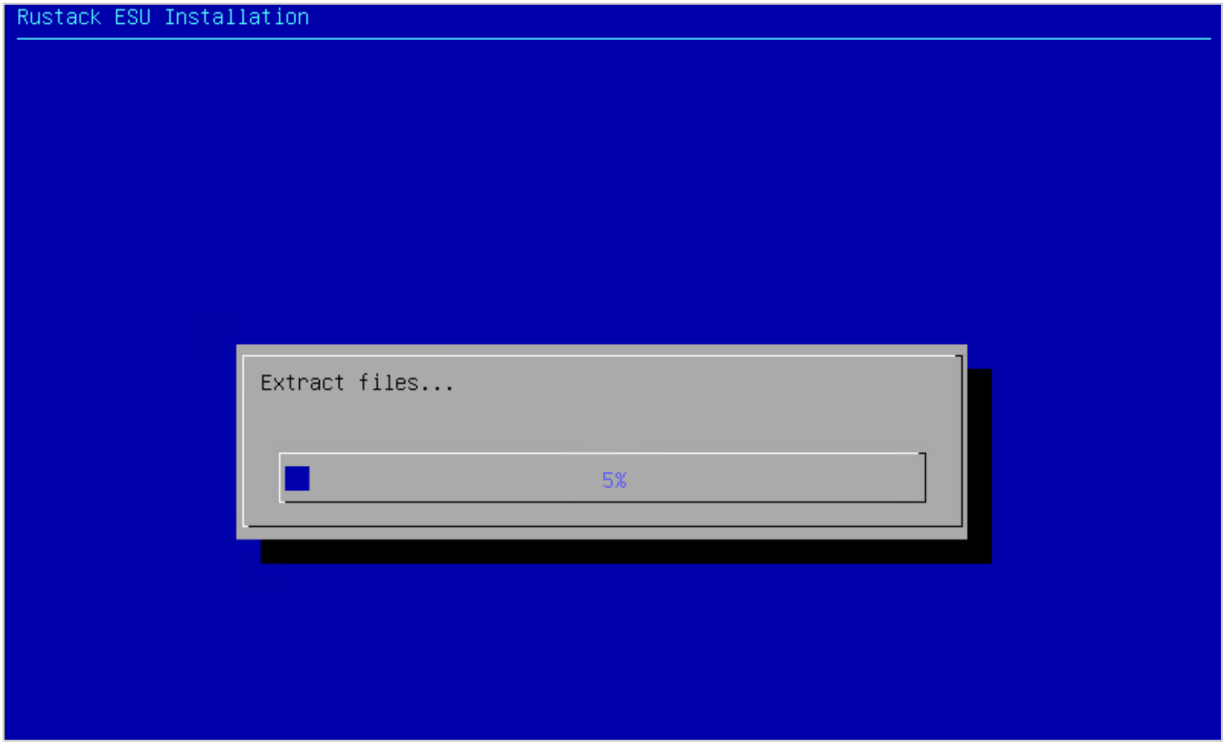

Рисунок 12

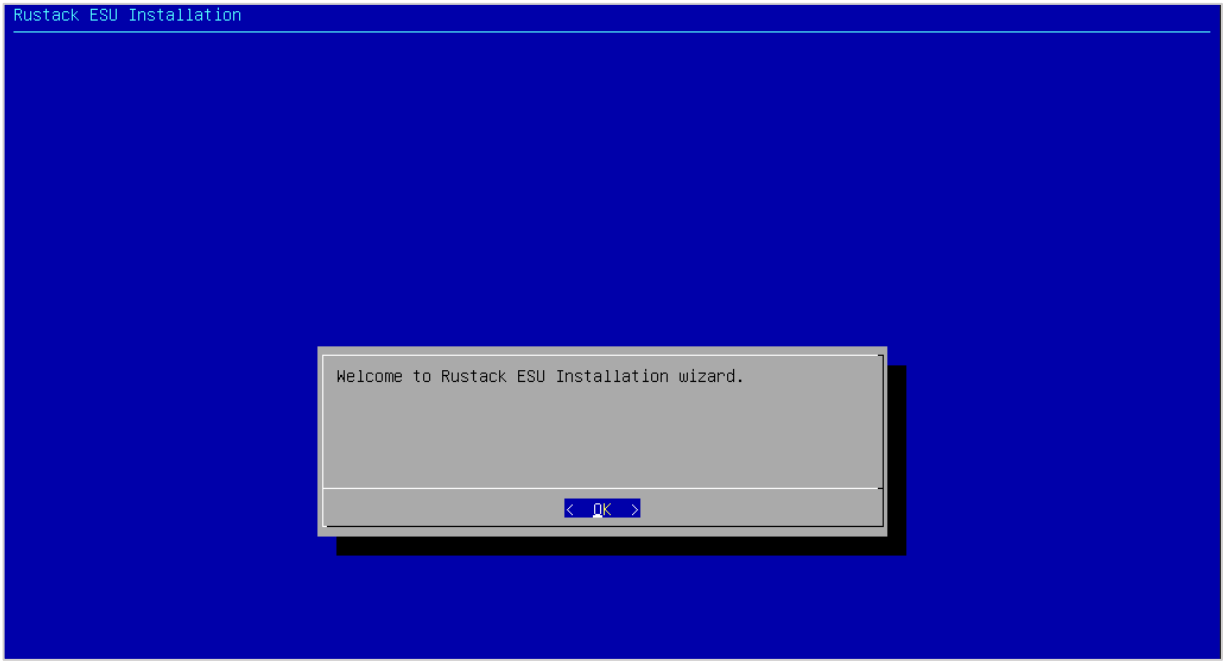

Рисунок 13

Далее будет задано несколько вопросов относительно сетевой конфигурации:

Сначала нужно указать какой IP был назначен ВМ ESU-box внутри РУСТЭК и какой подсетью будет располагать (***Рисунок 14*** и ***Рисунок 15***).

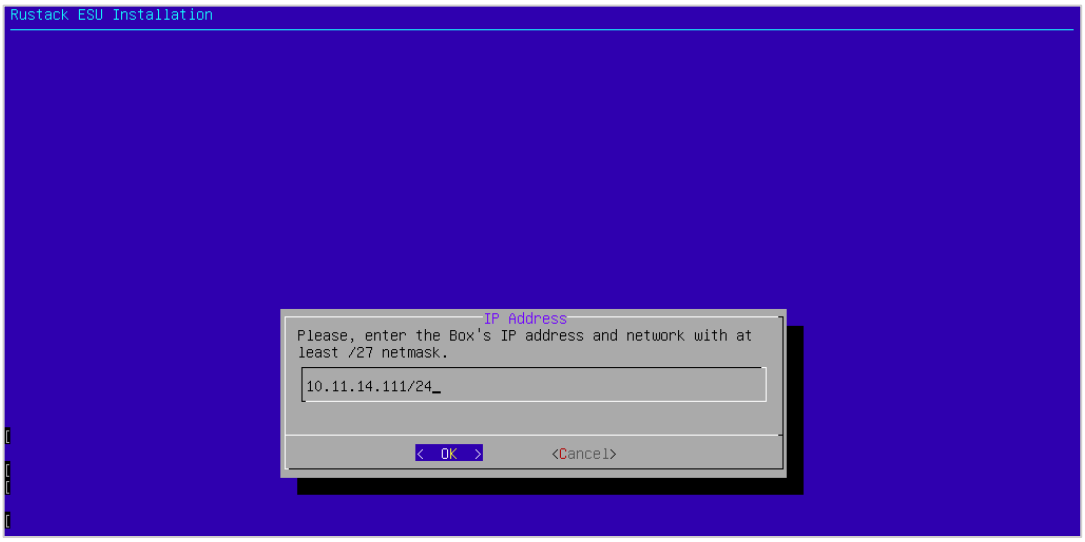

Рисунок 14

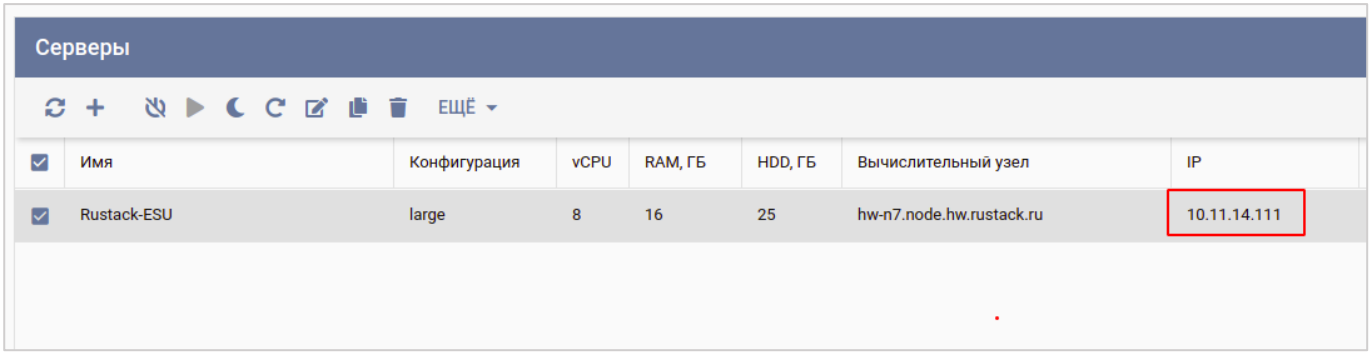

Рисунок 15

Далее необходимо ввести шлюз подсети (***Рисунок 16***).

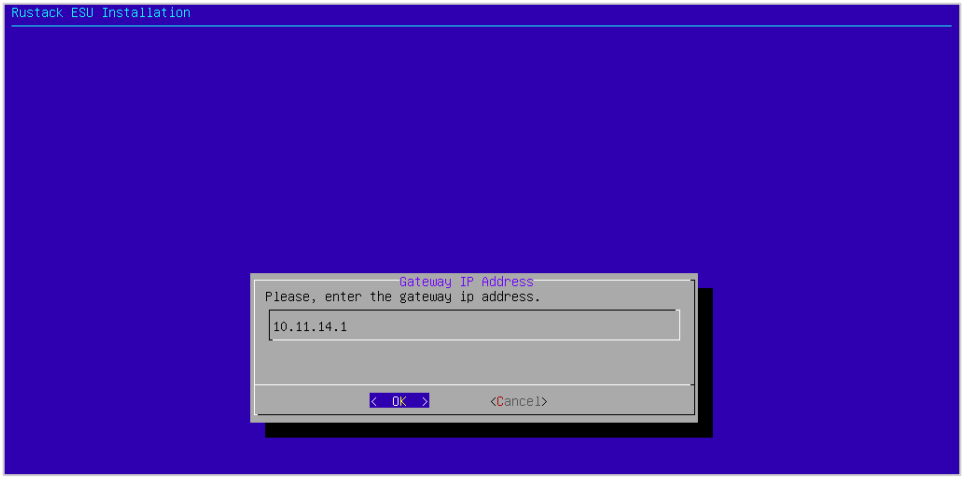

Рисунок 16

В следующем окне инсталлятора необходимо ввести VLAN ID, если на ESU-box подана сеть с несколькими VLAN. В нашем случае подан один VLAN, а значит данное поле заполнять не нужно (***Рисунок 17***).

Рисунок 17

Затем следует указать надо ли запускать в ESU-box DHCP-сервер? – Поскольку в нашей сети его нет, выбираем «Yes», используя кнопку пробел (***Рисунок 18***).

!!!Важно!!! Запуск DHCP-сервера на ESU-box обязателен.

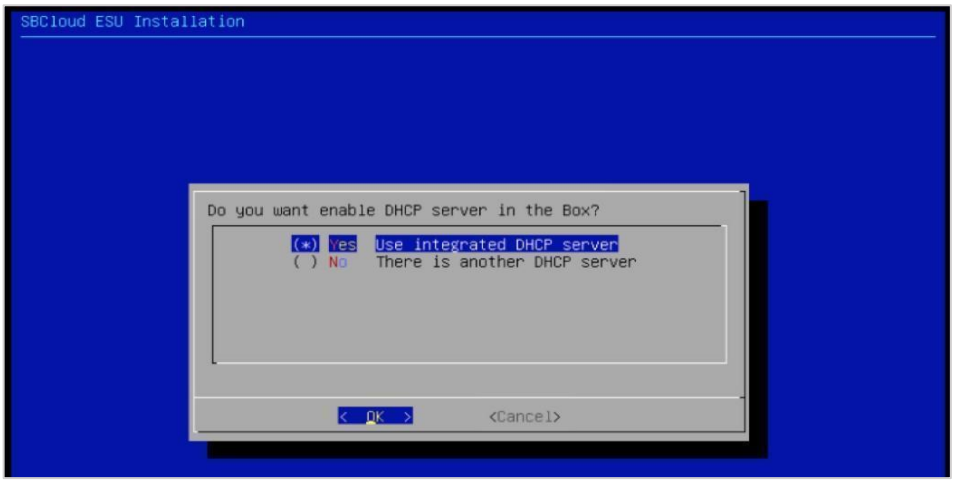

Рисунок 18

Затем нужно ввести адрес DNS-сервиса (***Рисунок 19***).

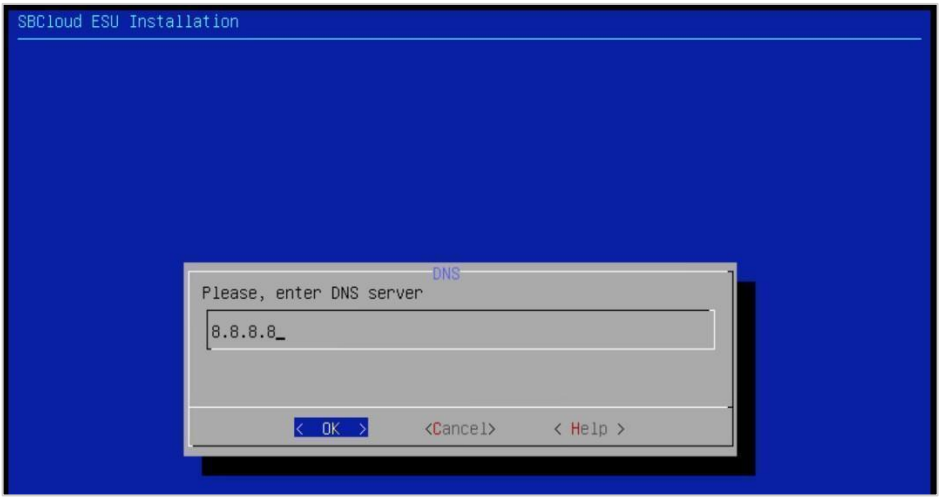

Рисунок 19

Далее нужно указать адрес SMTP-сервера. Он должен поддерживать подключение без авторизации. Можно оставить значение по умолчанию для использования встроенного SMTP-сервера (***Рисунок 20***).

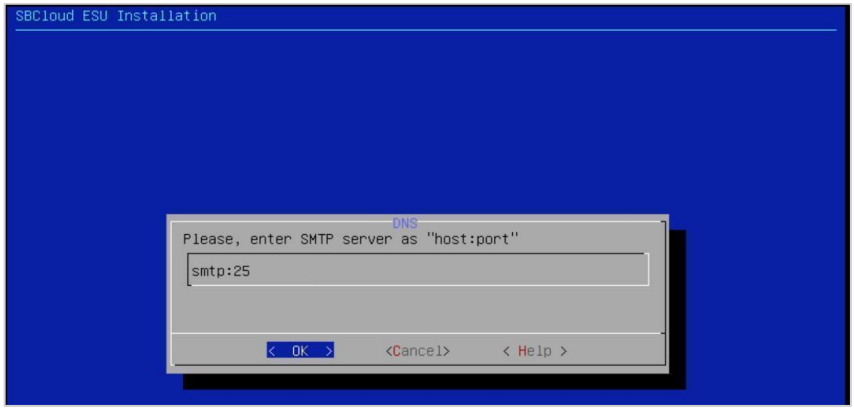

Рисунок 20

Теперь указываем пароль, который будет установлен для пользователя admin с правами администратора платформы (***Рисунок 21***).

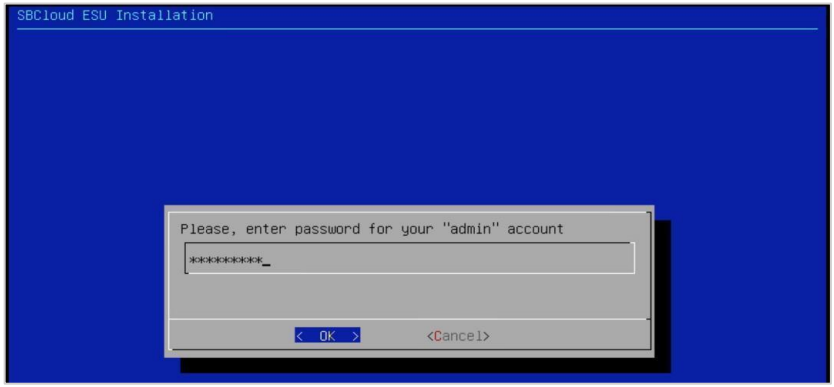

Рисунок 21

После всего этого необходимо дождаться завершения процесса настройки (***Рисунок 22*** – ***Рисунок 24***).

Рисунок 22

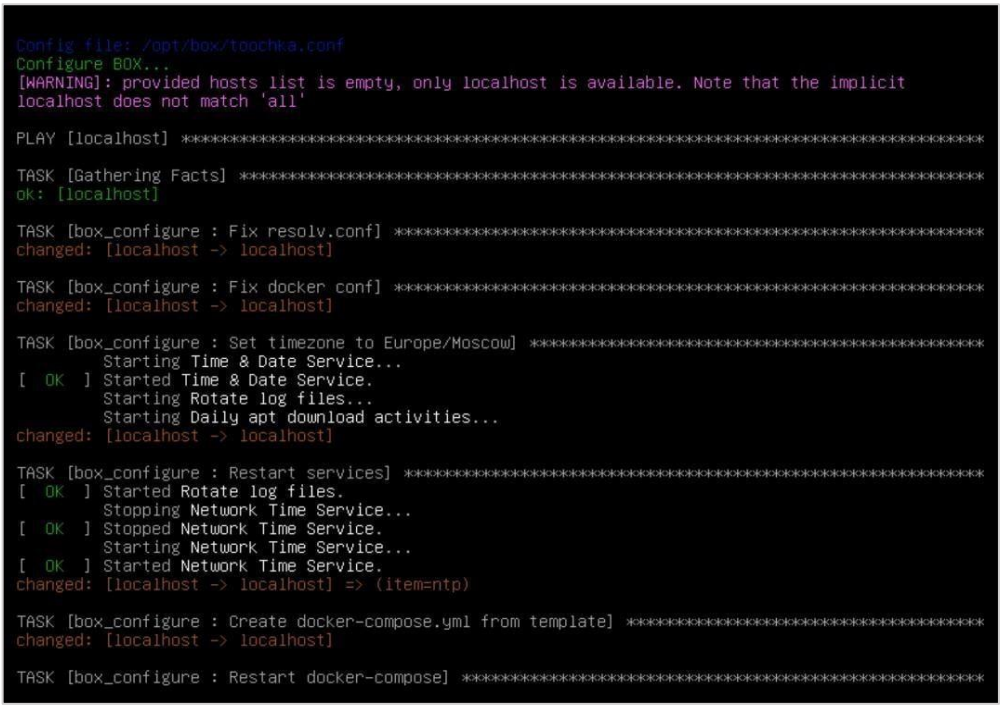

Рисунок 23

Рисунок 24

На этом установка РУСТЭК-ЕСУ завершена.

4. Настройка РУСТЭК-ЕСУ

После завершения установки, по IP адресу порта созданного сервера ESU-box (также указывался при инсталляции) будет доступен web-интерфейс РУСТЭК-ЕСУ. В нашем случае это https://10.11.14.111/cp (обратите внимание, что нужно использовать https://).

Авторизуйтесь с логином admin и паролем, заданным при инсталляции (***Рисунок 25***).

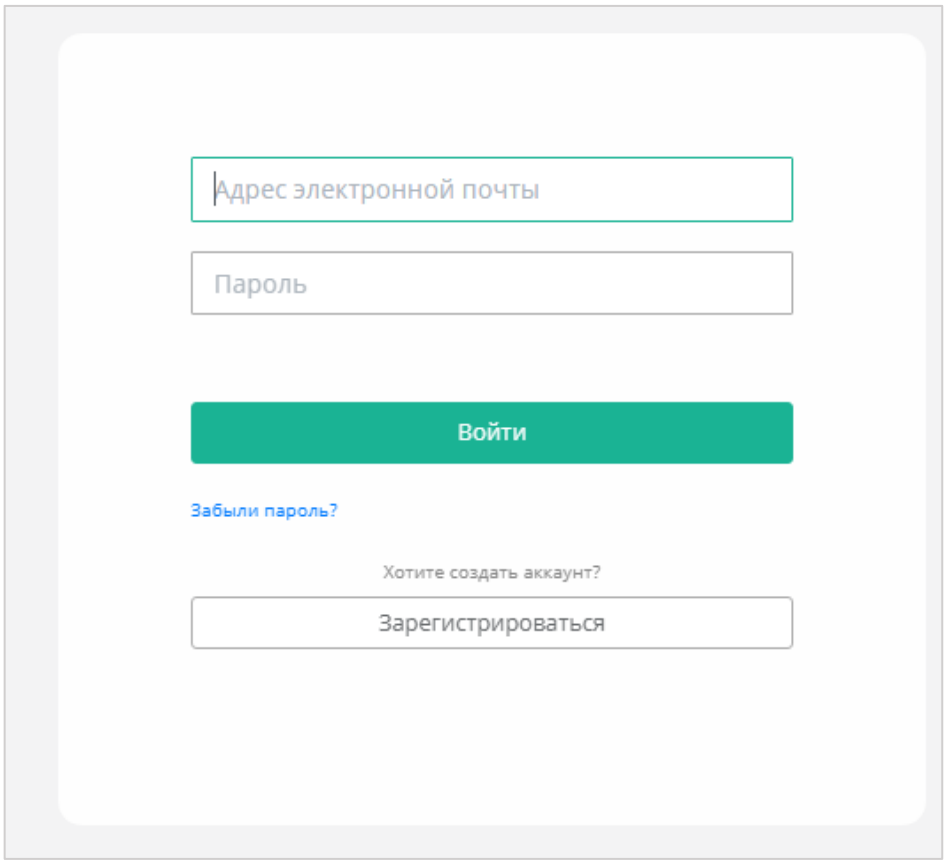

Рисунок 25

В меню **Администрирование** – **Тарифные планы** создается тарифный план – достаточно задать только название (***Рисунок 26***).

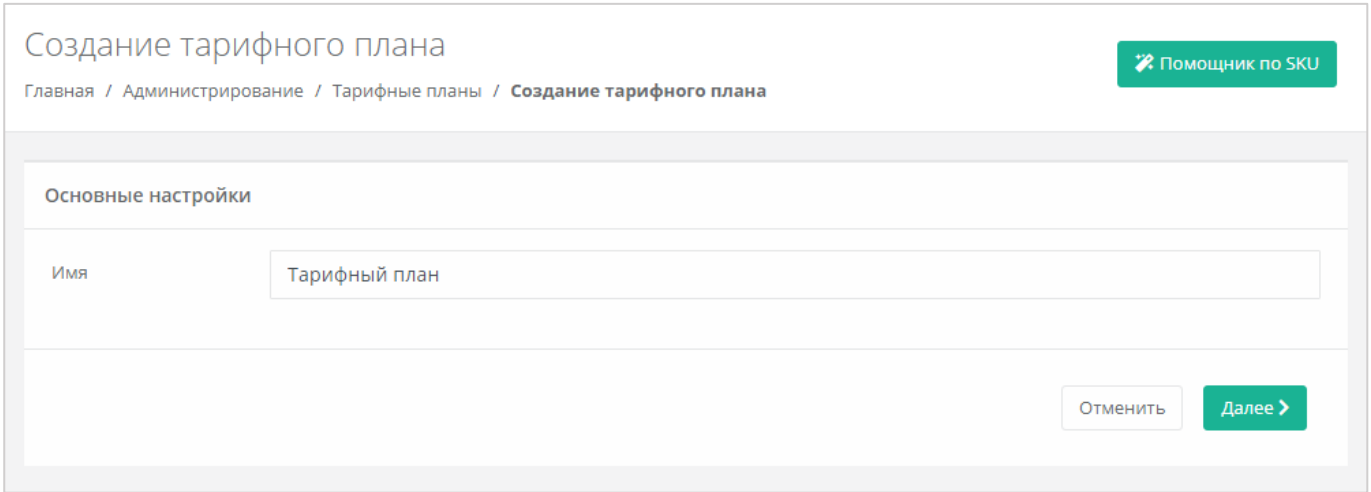

Рисунок 26

В меню **Администрирование** – **Партнёры**, создается партнёр – необходимо указать тарифный план (***Рисунок 27***).

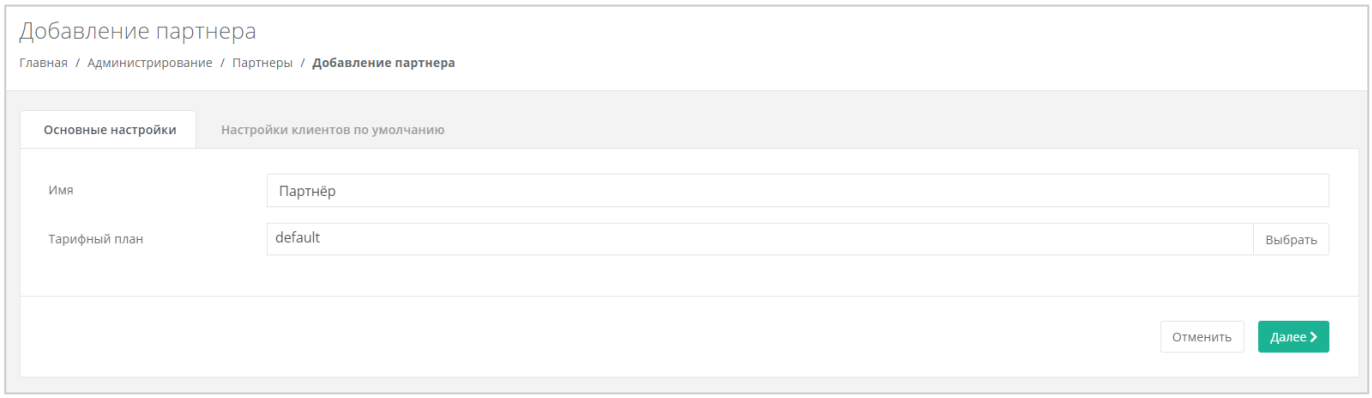

Рисунок 27

В меню **Администрирование** – **Домены**, партнёр привязывается к домену, и также производятся настройки домена – задаются/изменяются шаблоны писем (***Рисунок 28***).

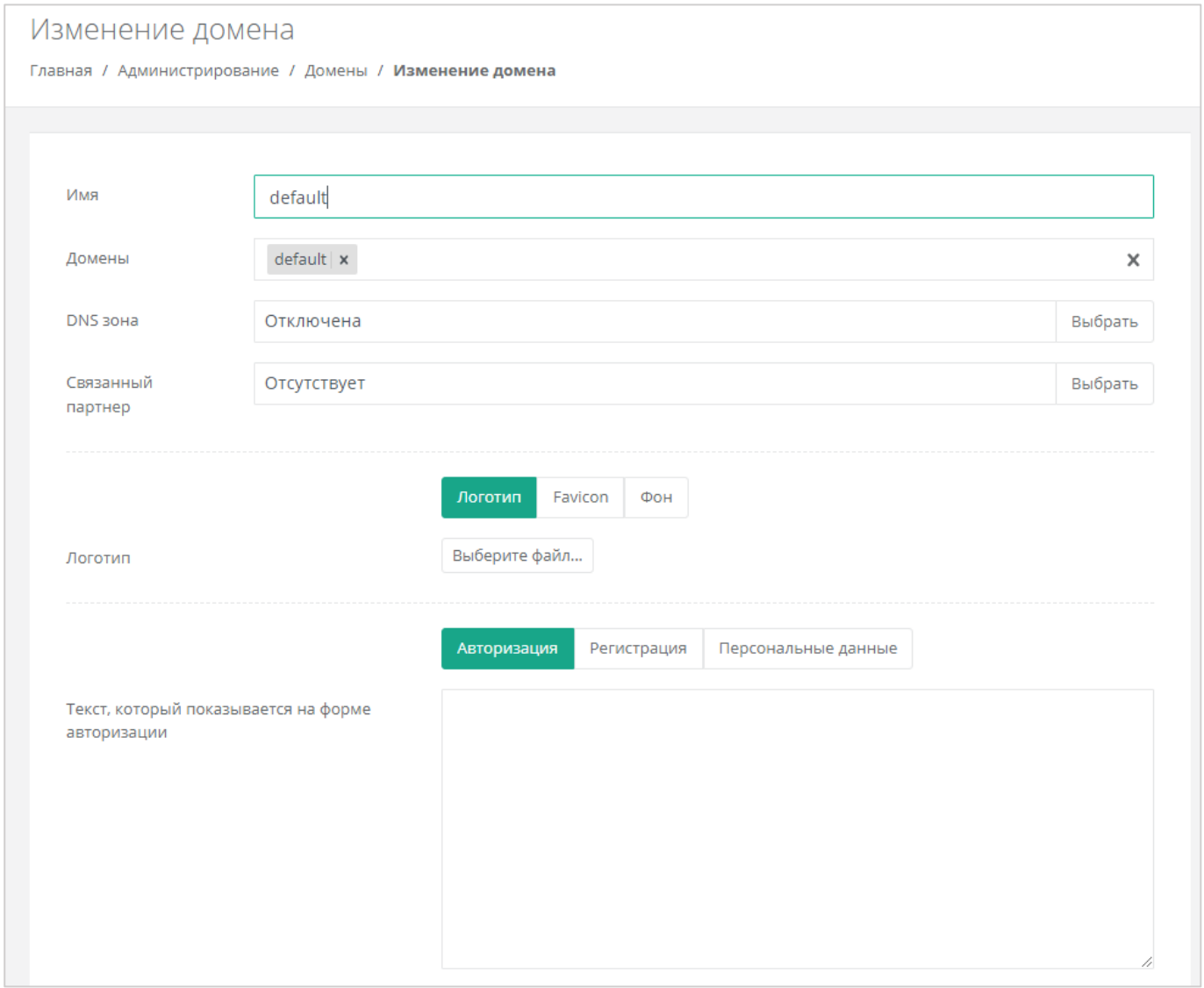

Рисунок 28

5. Настройка сегмента РУСТЭК/KVM

В случае если для управления РУСТЭК-ЕСУ необходимо добавить несколько инсталляций РУСТЭК (сегментов), то для каждого из них нужно выполнить все нижеперечисленные настройки.

5.1. Настройка сетевых зон для KVM сегмента

В меню **Инсталляция** – **Ресурсы** – **Сетевые зоны** настраивается сетевая зона (***Рисунок 29*** и ***Рисунок 30***).

Необходимо указать диапазон VLAN для пользовательских проектов (в данном случае – 701-1000).

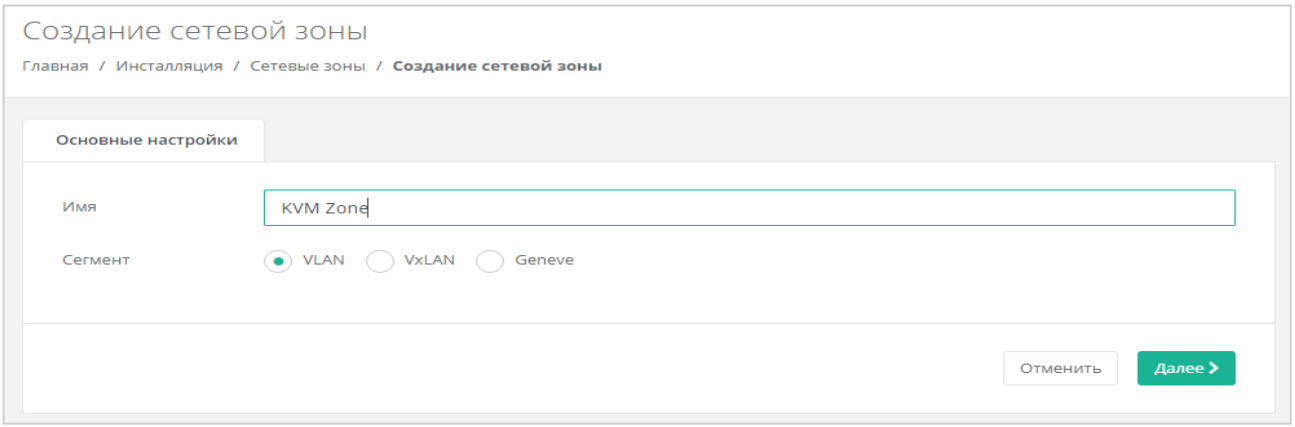

Рисунок 29

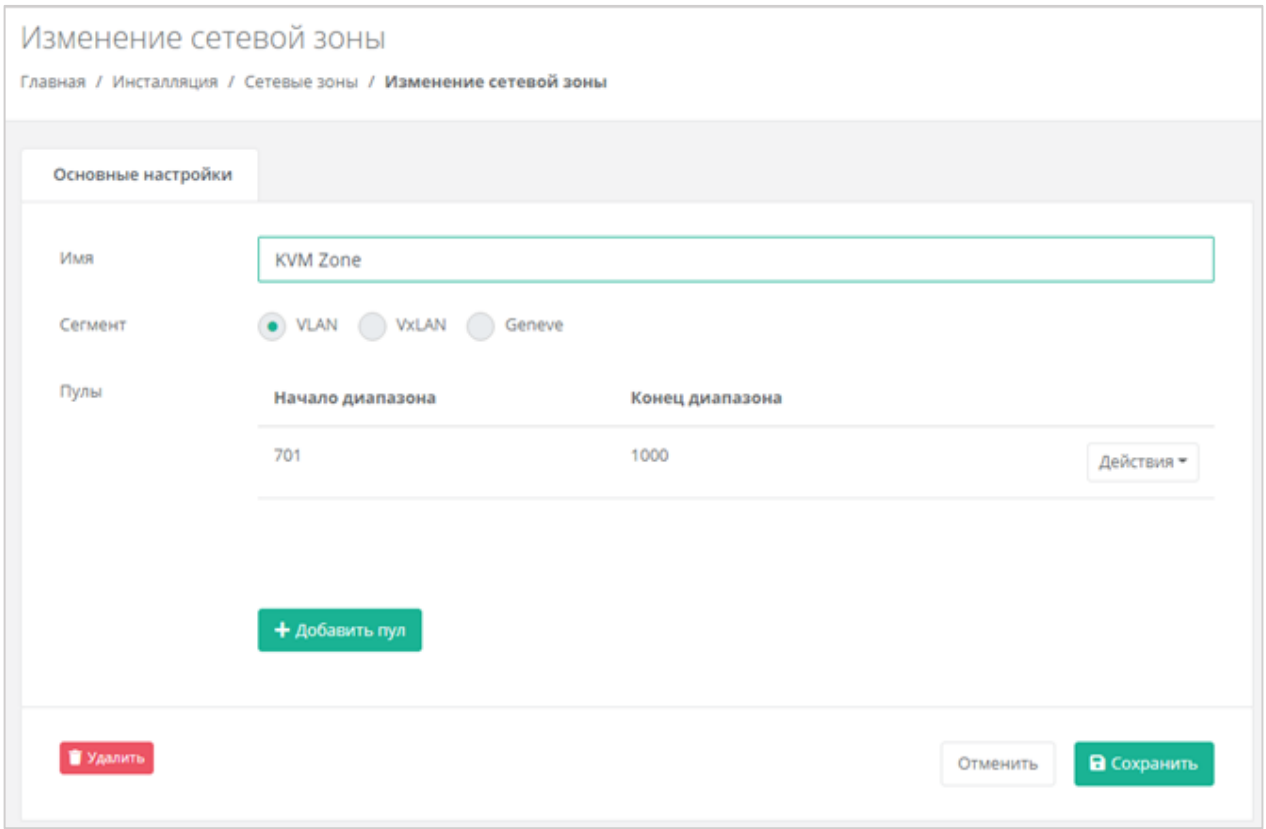

Рисунок 30

Аналогично создаём вторую сетевую зону для внешней сети (***Рисунок 31***).

Вводим диапазон для External-сети (один и тот же VLAN указывается и в начале, и в конце диапазона, в данном случае – 41).

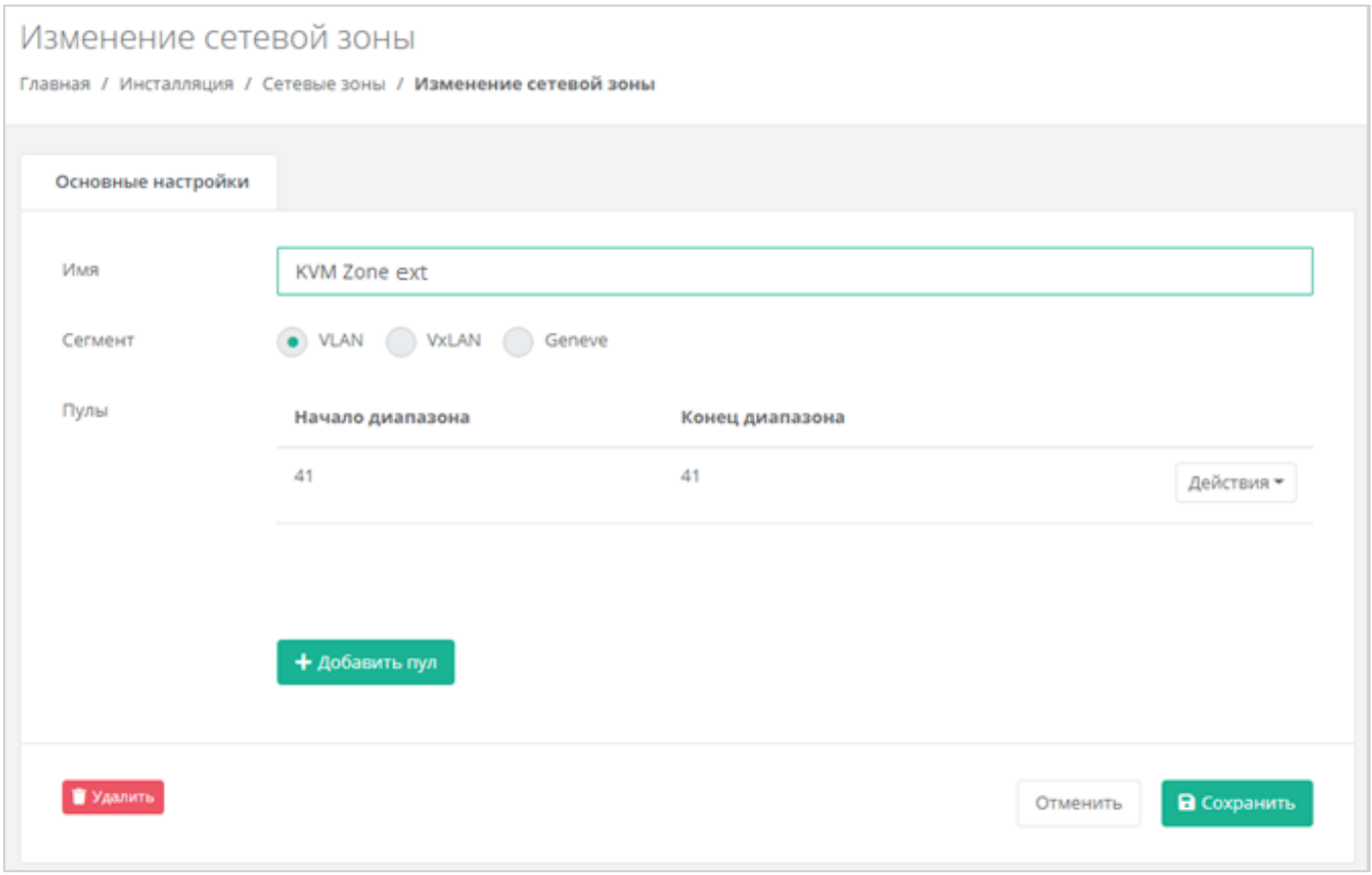

Рисунок 31

В меню **Инсталляция** – **Ресурсы** – **Сети и IP** создаем external-сеть (***Рисунок 32***).

- Название – любое.
- Сетевая зона – созданная ранее для внешней сети KVM-сегмента.
- VID/VNID – VLAN внешней сети (в нашем случае 41).

Указываем в настройках ранее созданную сетевую зону и соответствующий VLAN, указанный в сетевой зоне (в данном случае – 41).

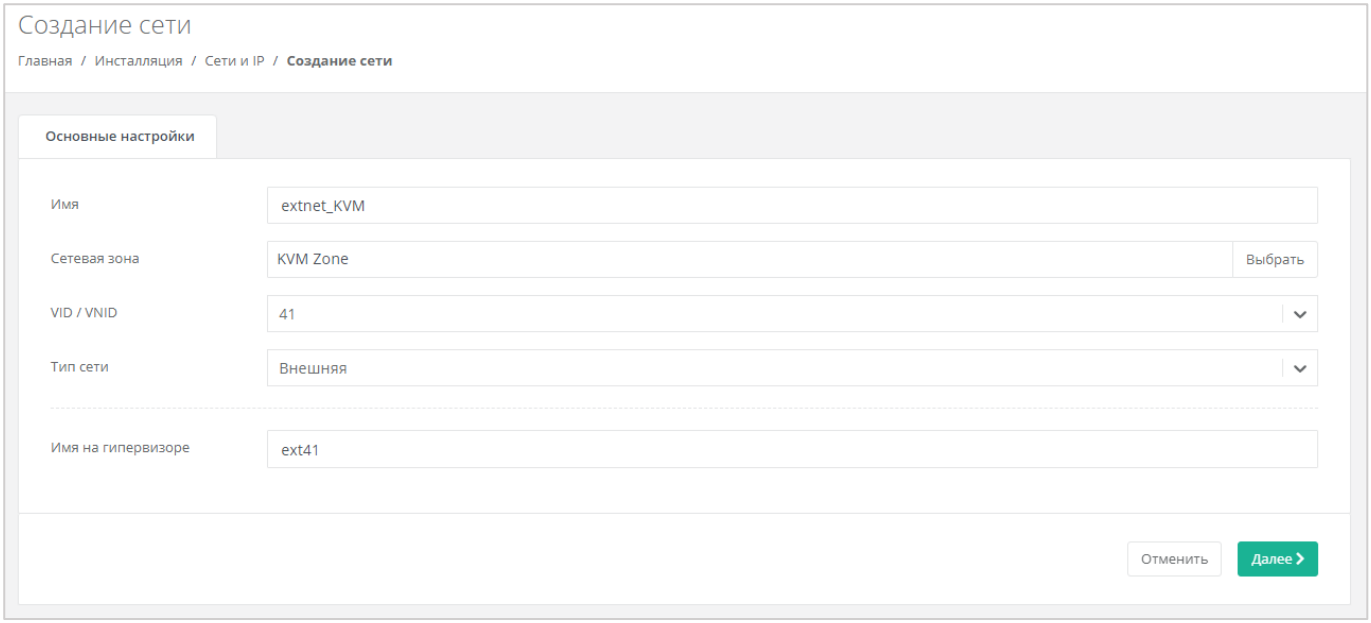

Рисунок 32

Добавляем подсеть с конфигурацией сети. DHCP должен быть **выключен**, CIDR надо указывать полный. Если нужно порезать диапазон выдаваемых IP, можно указать произвольный диапазон (***Рисунок 33***).

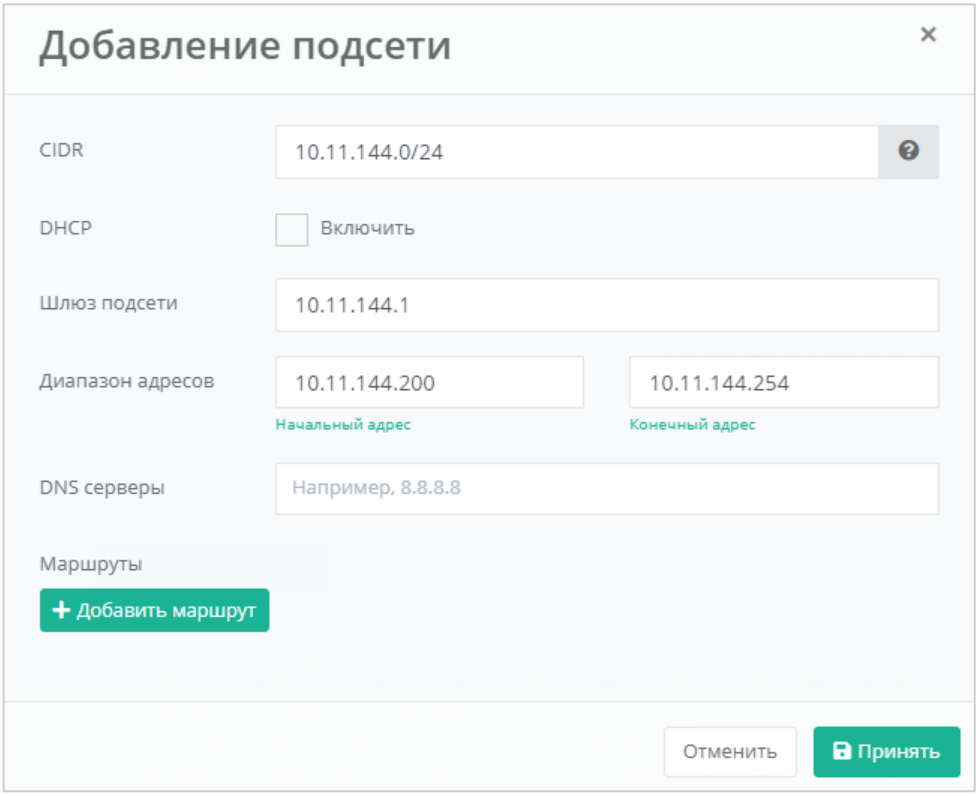

Рисунок 33

Данная внешняя сеть автоматически будет создана при первом создании ВЦОД в KVM.

5.2. Настройка Openstack-раннера

Следующим этапом в меню **Инсталляция** – **Система** – **Раннеры** конфигурируется OpenStack-раннер для KVM-сегмента. Указывается IP-адрес РУСТЭК, логин администратора и пароль для авторизации посредством Keystone и Содержимое файла clouds.yml, который находится по следующему пути: /etc/openstack/clouds.yml на контроллер-ноде РУСТЭК (***Рисунок 34***).

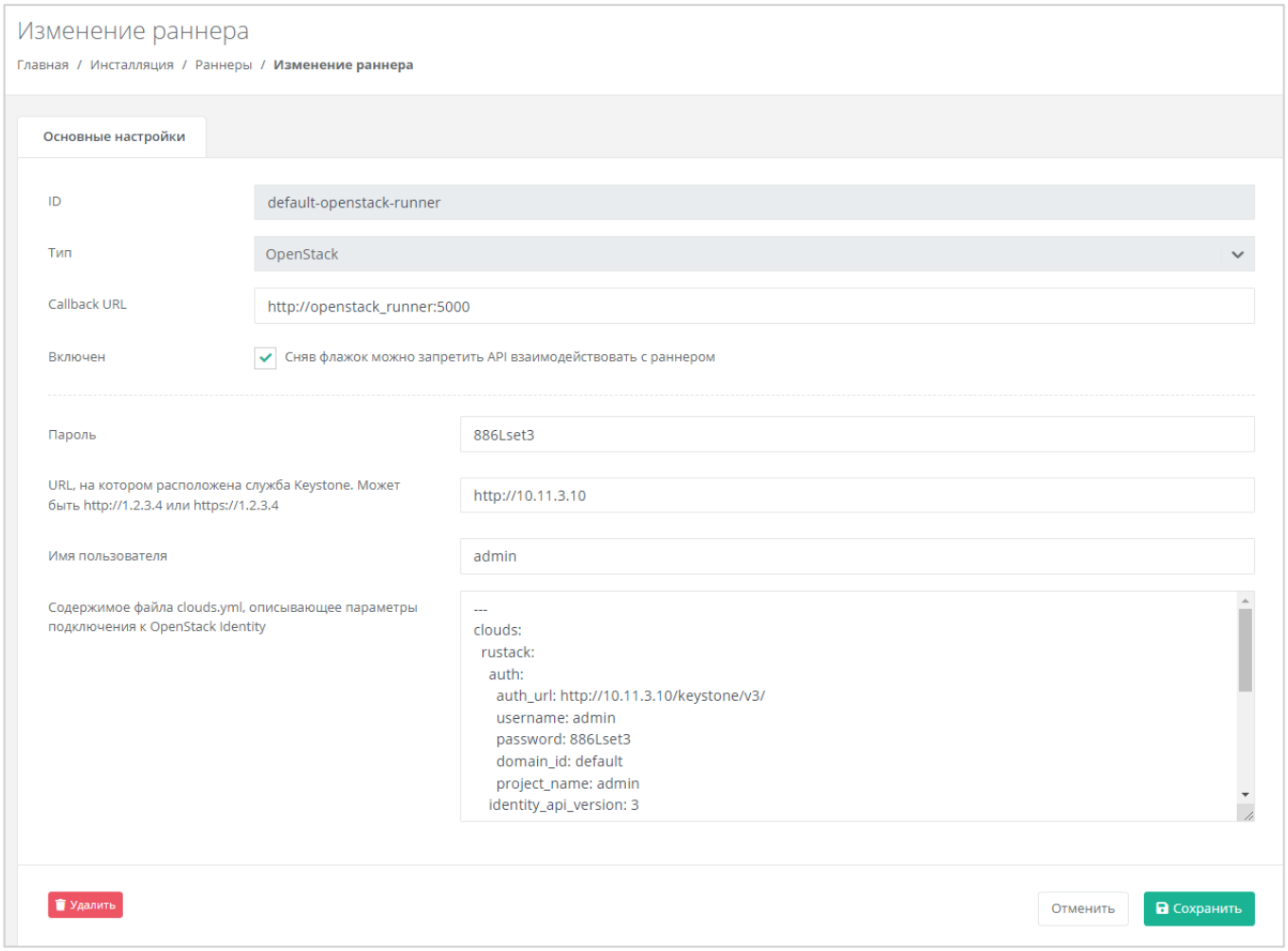

Рисунок 34

Если настройки произведены правильно, то индикатор OpenStack-раннера должен быть зелёным (***Рисунок 35***).

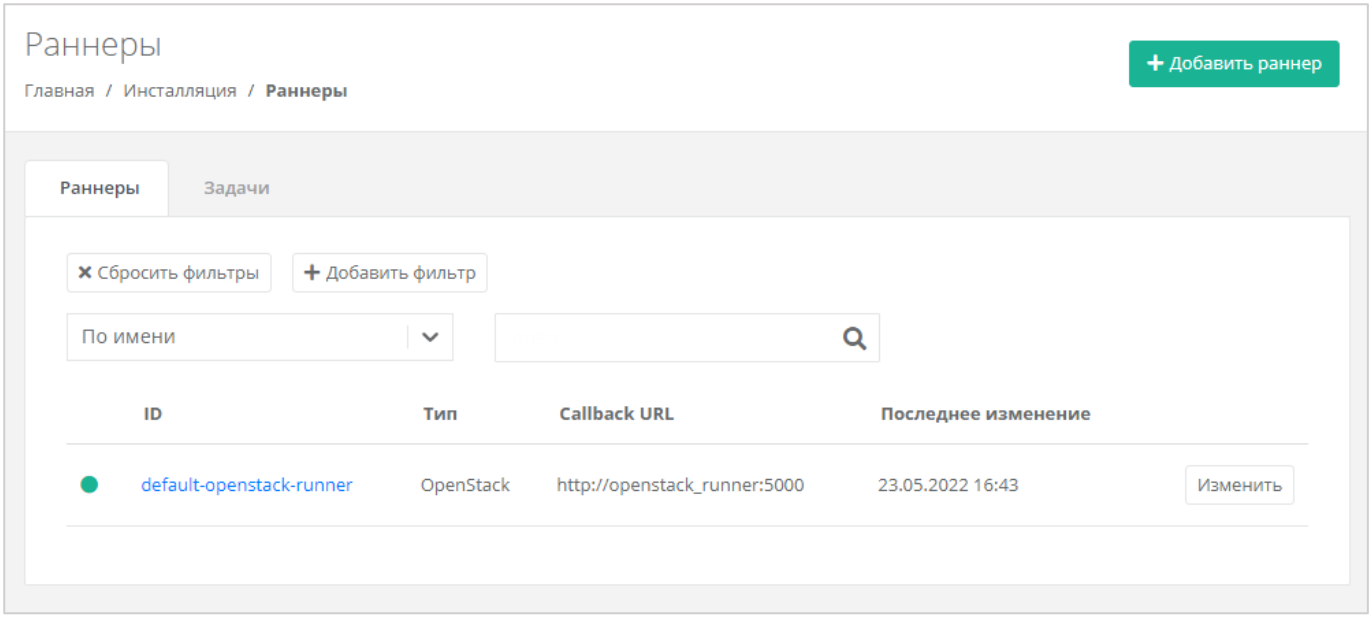

Рисунок 35

5.3. Настройка ресурсного пула для KVM-сегмента

Далее в меню **Инсталляция** – **Ресурсы** – **Ресурсные пулы**, конфигурируется ресурсный пул – необходимо указать соответствующий раннер (в данном случае – *default-openstack-runner*), сетевую зону и внешнюю сеть (***Рисунок 36***).

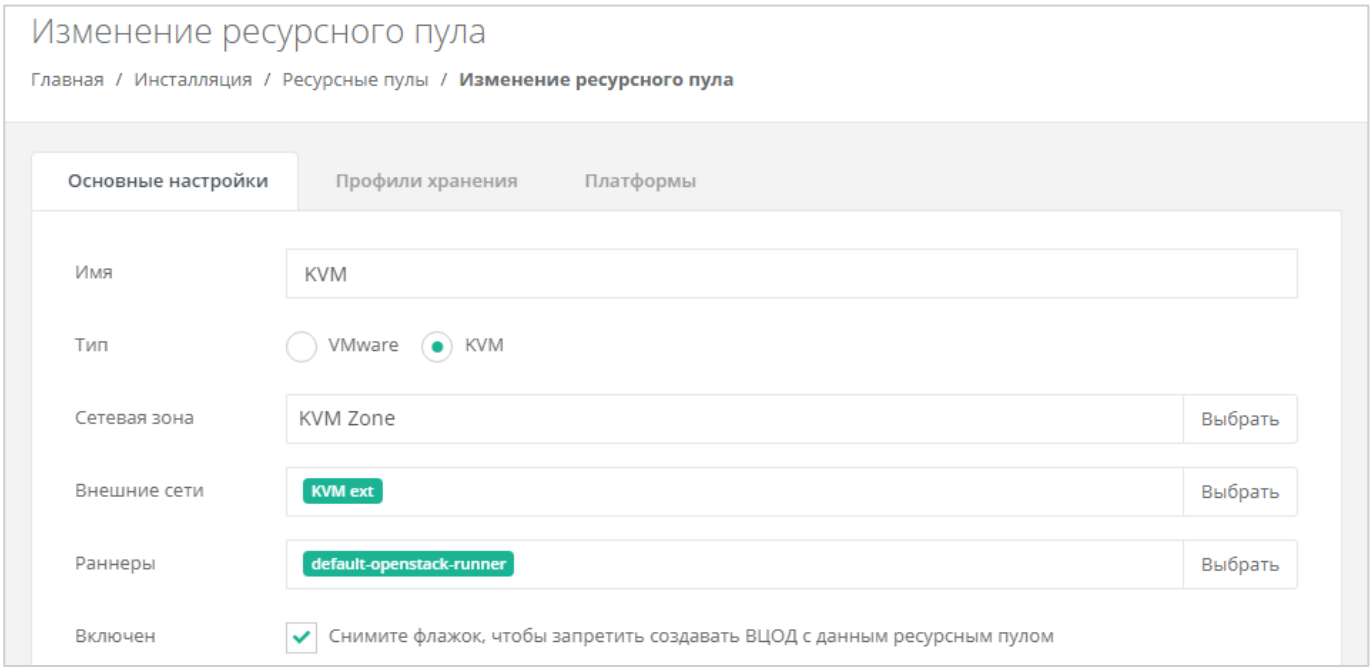

Рисунок 36

Далее необходимо настроить переподписку vCPU и RAM (***Рисунок 37***).

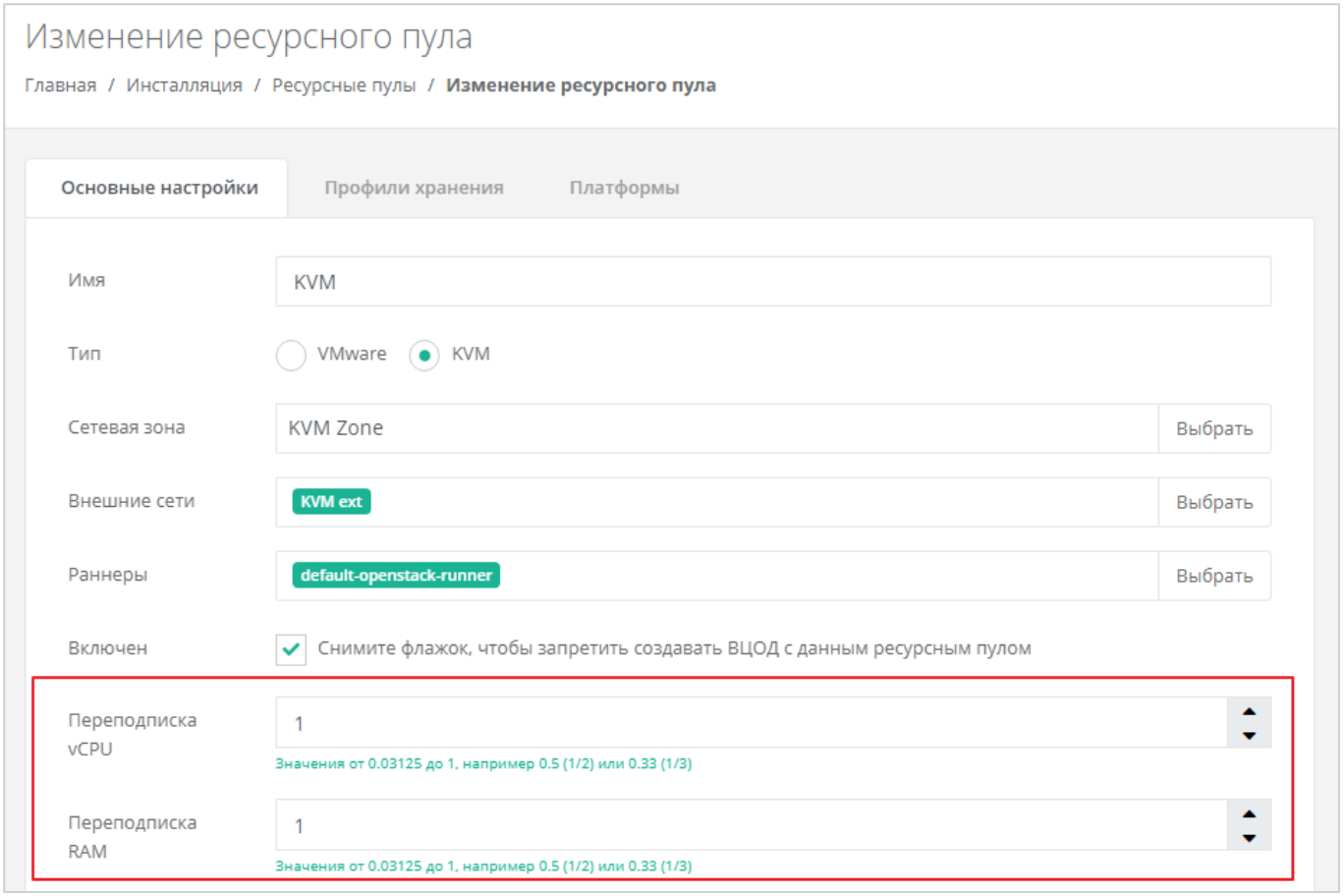

Рисунок 37

После этого необходимо настроить ограничения на один сервер (***Рисунок 38***)

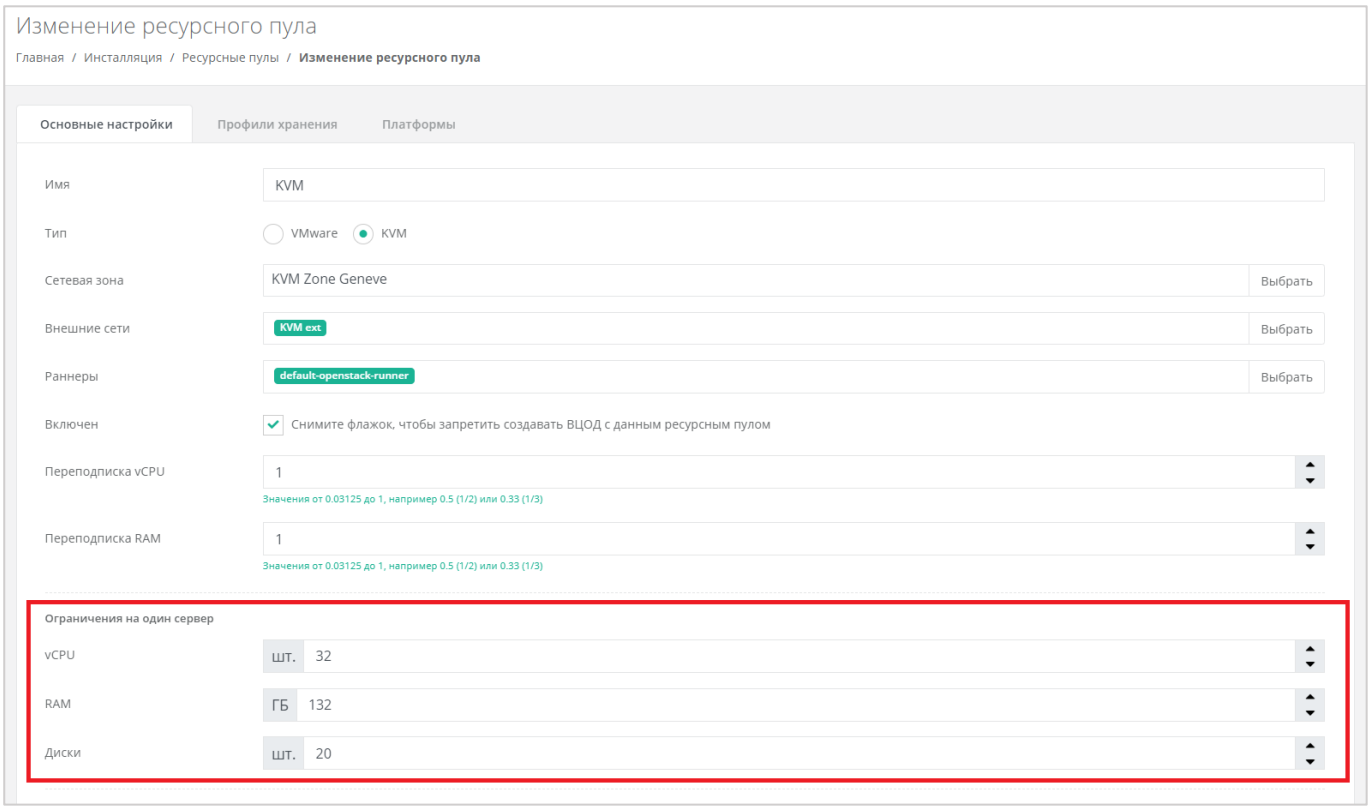

Рисунок 38

Для KVM-сегмента заполнять поля ниже не нужно.

После сохранения новых настроек ресурсного пула РУСТЭК-ЕСУ должна забрать адреса сервисных портов РУСТЭК в свою БД. В этом можно убедиться, запустив в консоли ВМ ESU-box команду:

```
sudo docker-compose exec api make shell
```

В открывшейся консоли ввести:

```
Port.objects.table('id', 'type', 'network_id', 'ip_address')
```

Появится табличная форма аналогично представленной ниже (***Рисунок 39***).

```
In [1]: Port.objects.table('id', 'type', 'network_id', 'ip_address')
   ...:
+--------------------------------------+---------+--------------------------------------+---------------+
| id                                   | type    | network_id                           | ip_address    |
|--------------------------------------+---------+--------------------------------------+---------------|
| a1444fb8-5e72-4c9e-af43-f6ff8474f1a2 | service | 3c31f9fc-3e5f-43b5-aaa2-27b434b38917 | 185.17.143.89 |
| 884627df-feaa-4b44-b859-1fe00317726b | service | 3c31f9fc-3e5f-43b5-aaa2-27b434b38917 | 185.17.143.75 |
| 6db921b9-91f6-4bbd-b108-ca2cf20588e8 | service | 3c31f9fc-3e5f-43b5-aaa2-27b434b38917 | 185.17.143.76 |
| a47b37ac-bc68-4b4e-803d-48b105334256 | service | 3c31f9fc-3e5f-43b5-aaa2-27b434b38917 | 185.17.143.87 |
+--------------------------------------+---------+--------------------------------------+---------------+
```

Рисунок 39

Количество записей в таблице может отличаться в зависимости от инсталляции, но таблица не должна быть пустой. **В случае, если таблица пуста, проверьте, не была ли допущена ошибка в названии external-сети – она должна называться openstack network list-external.**

Далее конфигурируются профили хранения ресурсного пула (***Рисунок 40*).**

- Имя – в соответствии с подсказкой (SSD, SATA, SAS).
- Отображаемое имя – любое.
- Имя вольюм тайпа – в соответствии с доступными volume type в РУСТЭК.
- Биллинг-класс – выбираем необходимый.

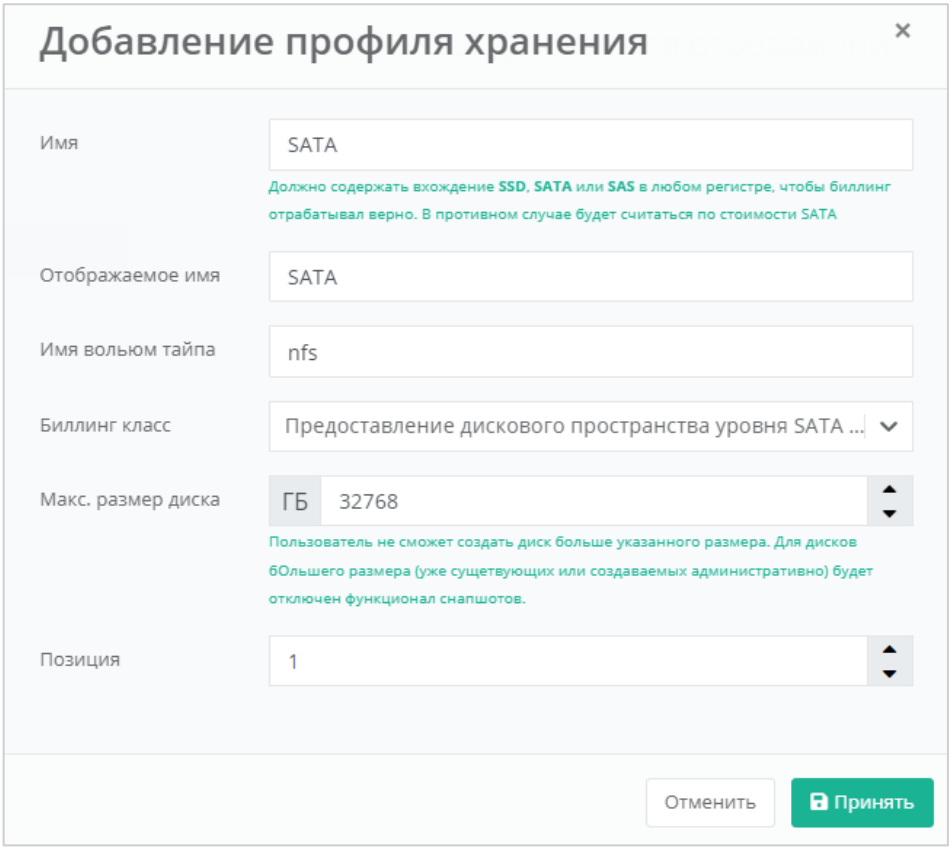

Рисунок 40

Имя volume type в РУСТЭК можно получить, выполнив на одном из контроллеров РУСТЭК команду:

```
openstack volume type list --public
```

Будет выведен приблизительно следующий список (***Рисунок 41***):

```
aio ~ # openstack volume type list --public
+--------------------------------------+-------------+-----------+
| ID                                   | Name        | Is Public |
+--------------------------------------+-------------+-----------+
| c5c47b8e-352c-42ba-94af-9116bf5fb886 | nfs         | True      |
| 77110d5f-0b96-45bf-9df5-65d87df4ed76 | __DEFAULT__ | True      |
+--------------------------------------+-------------+-----------+
```

Рисунок 41

В качестве вольюм тайпа в панели управления РУСТЭК-ЕСУ необходимо указать значение поля «Name». В нашем случае это nfs.

Далее необходимо проверить правильность заполнения вкладки «Платформа». Если настройки отсутствуют или не совпадают – нужно нажать кнопку «Добавить платформу» и в открывшемся окне указать имя агрегата из РУСТЭК.

Список агрегатов можно получить, выполнив на одном из контроллеров РУСТЭК команду:

```
OS_CLOUD=rustack_system openstack aggregate list
```

Будет выведен приблизительно следующий список (***Рисунок 42***):

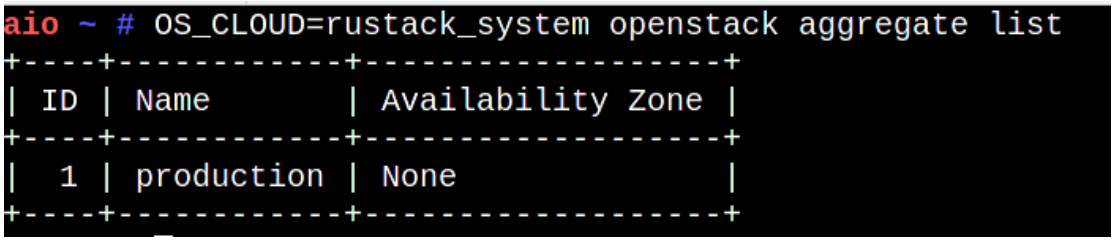
aio ~ # OS_CLOUD=rustack_system openstack aggregate list
+----+------------+-------------------+
| ID | Name | Availability Zone |
+----+------------+-------------------+
| 1 | production | None |
+----+------------+-------------------+

Рисунок 42

6. Создание шаблонов ВМ для сегмента РУСТЭК/KVM

Для создания шаблона ВМ необходим образ ОС с cloud-init. На сайте OpenStack есть ссылки для скачивания таких образов: https://docs.openstack.org/image-guide/obtain-images.html

Далее будет рассмотрен пример создания шаблона ВМ с операционной системой Ubuntu 18.04 LTS.

Заходим по SSH (root:rustack) на один из контроллеров РУСТЭК и скачиваем целевой образ, после чего создаем образ в РУСТЭК:

```
curl https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.img --output bionic-server-cloudimg-amd64.img


openstack image create --public --disk-format qcow2 --container-format bare --property distro=Ubuntu --property hw_disk_bus=scsi --property hw_scsi_model=virtio-scsi --property hw_vif_model=virtio --property image_type=master --file bionic-server-cloudimg-amd64.img --min-disk 10 --min-ram 2048 Ubuntu-Bionic-ESU3
```

Последний параметр команды – имя образа в РУСТЭК, его необходимо записать или запомнить.

Создаём шаблон в РУСТЭК-ЕСУ через веб-интерфейс, для этого переходим в меню **Инсталляция** – **Шаблоны** – **Серверы** и нажимаем «Создать шаблон», после чего открывается форма где необходимо указать сегмент (в данном случае – KVM), название шаблона, группу (***Рисунок 43***).

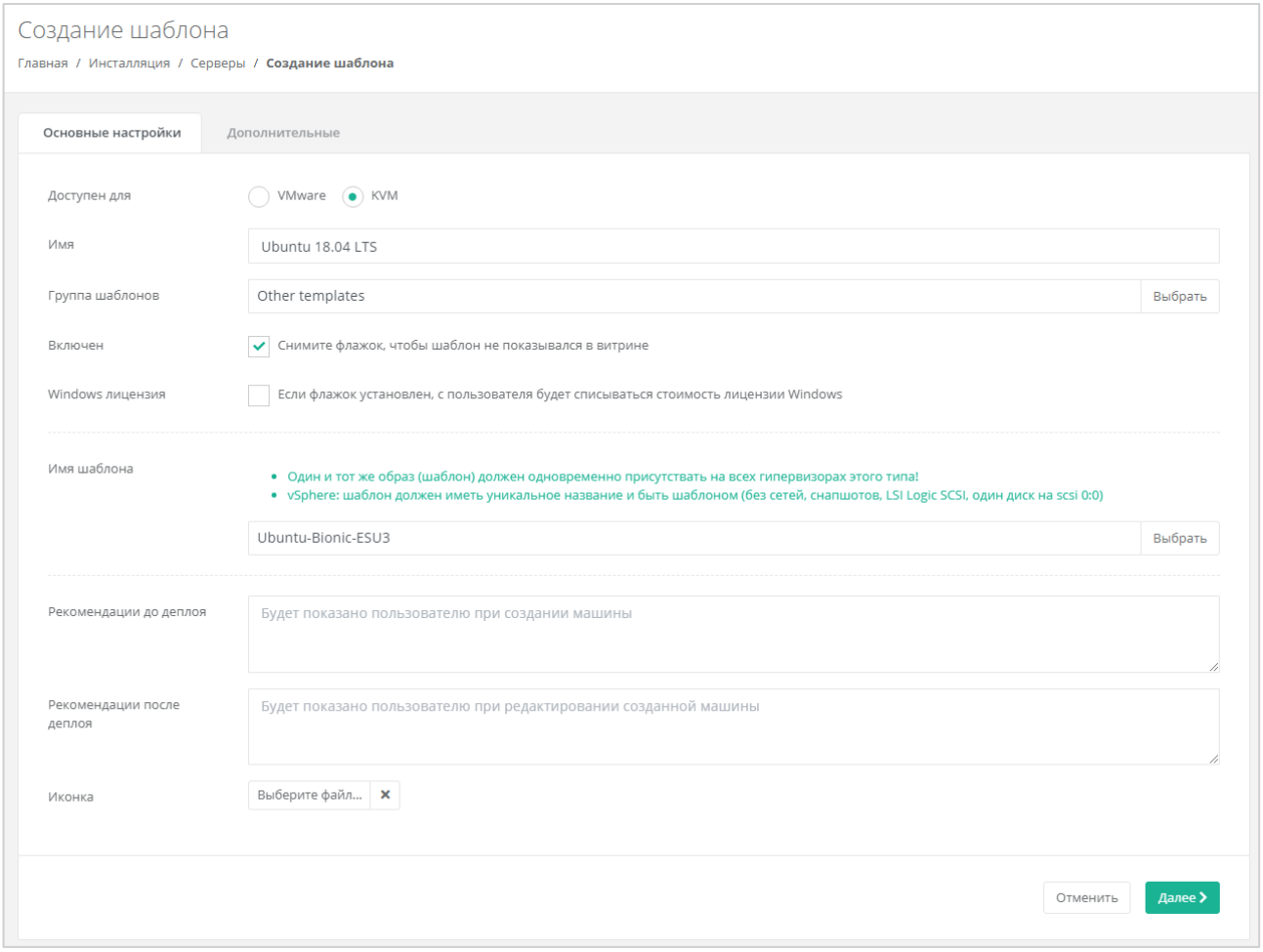

Рисунок 43

Нажимаем на имя шаблона – откроется список образов РУСТЭК, в котором необходимо выбрать ранее созданный образ (***Рисунок 44***).

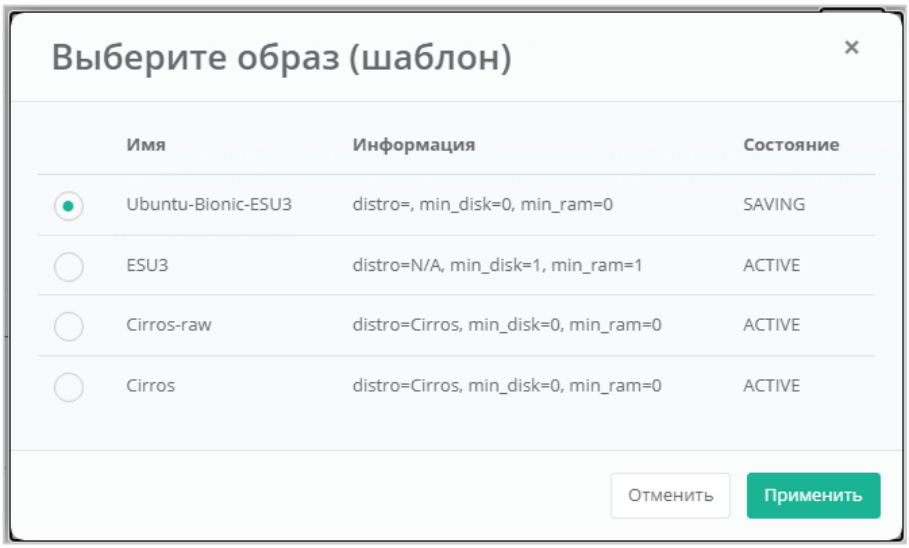

Рисунок 44

Указываем минимальное число ядер CPU (минимум 1 ядро) и объем RAM (минимум 2 Гб – ***Рисунок 45***).

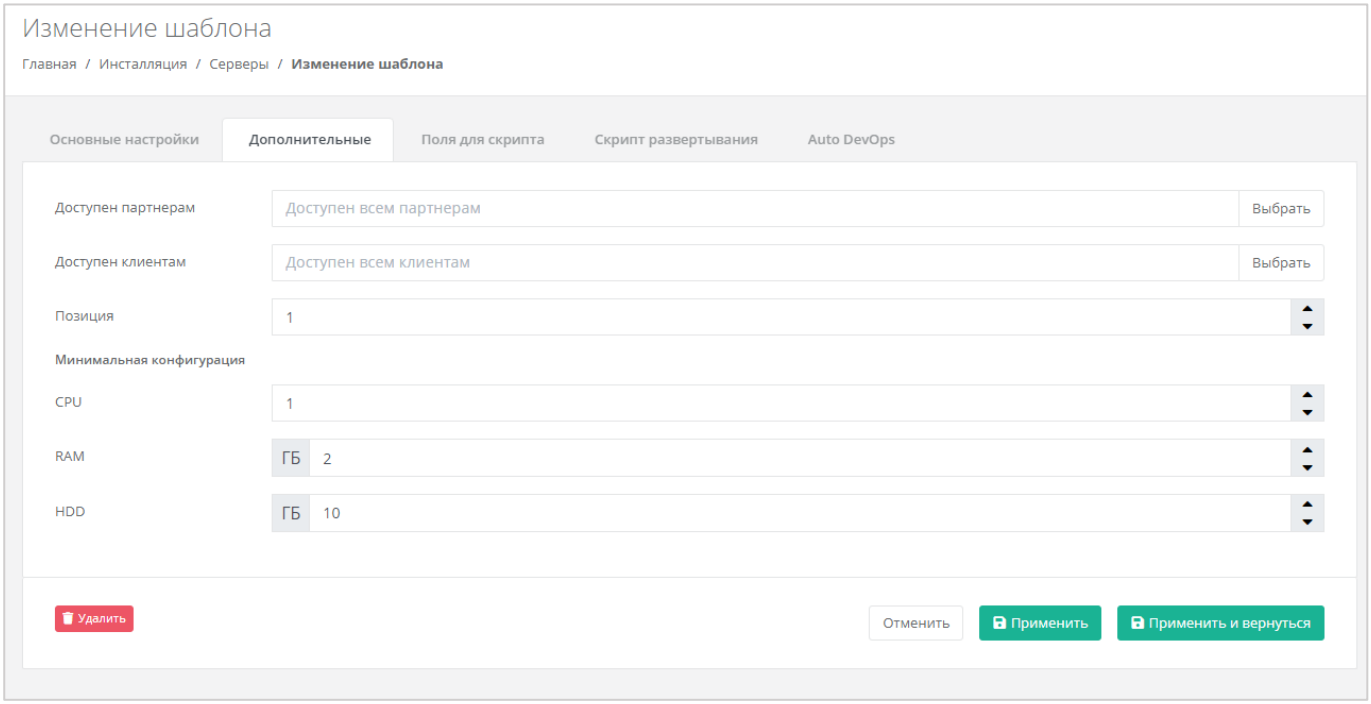

Рисунок 45

Переходим во вкладку поля для скрипта. Рекомендуем заполнить поля, указанные на скриншоте. (***Рисунок 46***).

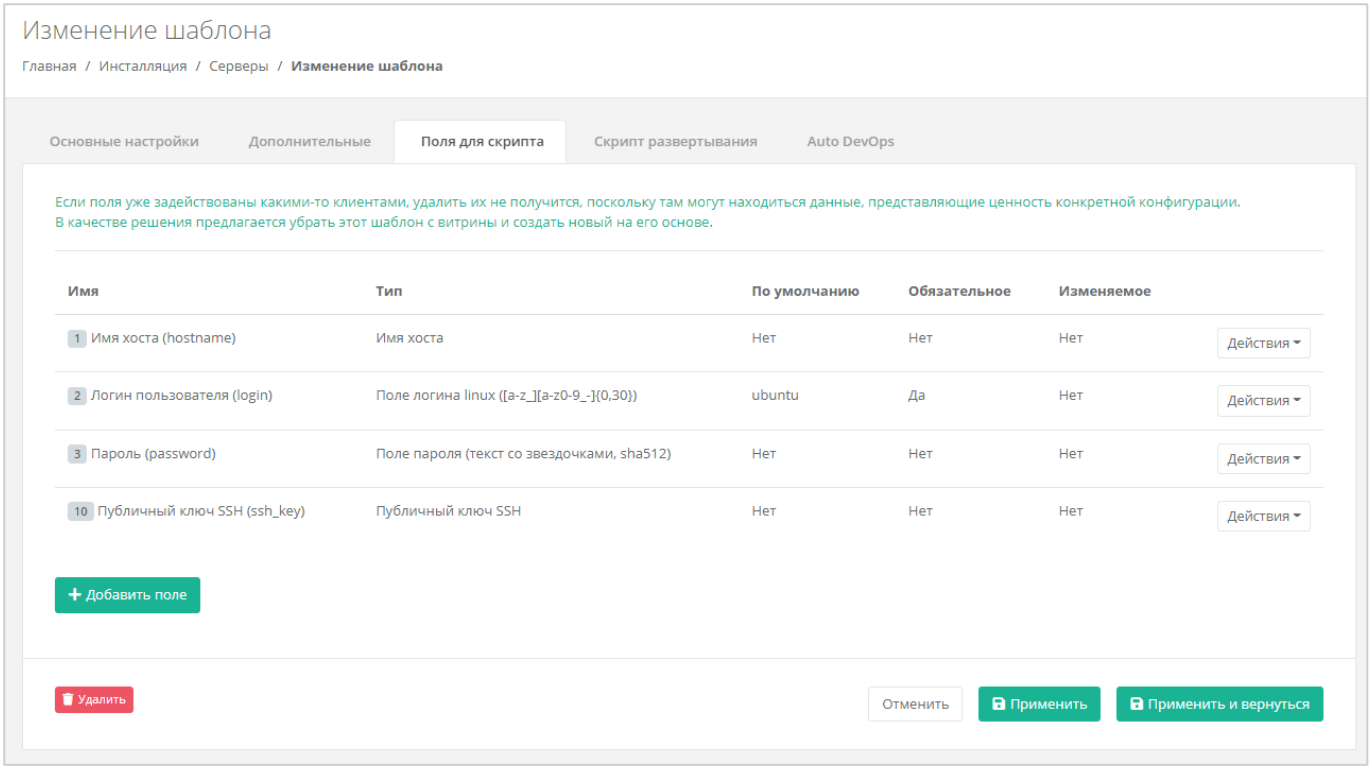

Рисунок 46

Далее во вкладке ***Скрипт развёртывания*** необходимо добавить скрипт развёртывания.

Скрипт развёртывания применяется во время развёртывания виртуальной машины внутри операционной системы сервера.

Примечание: универсальный скрипт развёртывания для Linux OS приложен ниже в документации в разделе «Универсальный скрипт развёртывания».

На вкладке ***Auto DevOps*** можно настроить Auto DevOps-скрипт. Скрипт обращается к API РУСТЭК-ЕСУ для выполнения указанных в скрипте операций.

Auto DevOps-скрипт пишется на языке Python и используется для выполнения дополнительных операций с сервером во время его создания и/или запуска.

Примечание: внесение изменений в Auto DevOps-скрипт рекомендуется только для вендоров. Просьба не редактировать настройки скрипта самостоятельно.

Пример скрипта приведён в Приложении 1.

!!!Важно!!! После внесения изменений в скрипт нужно обязательно нажать кнопку Применить.

В результате редактирования настроек Auto DevOps-скрипта вносятся изменения в панели управления. Например, применяются необходимые шаблоны брандмауэра после разворачивания виртуальной машины.

После внесения изменений нажимаем кнопку **Применить и вернуться**. Созданный шаблон ВМ появится в списке шаблонов и из него можно будет создавать ВМ.

7. Настройка сегмента VMware vSphere

В случае если для управления РУСТЭК-ЕСУ необходимо добавить несколько инсталляций VMware vSphere (сегментов), то для каждого из них нужно выполнить все нижеперечисленные настройки.

Необходимые работы на стороне VMware для подключения к РУСТЭК-ЕСУ:

1. Создать пользователя esu-admin с правами администратора.
2. Создать Datacenter.
3. Создать кластером хоста(ов) в Datacenter, внутри которого будут создаваться ВМ и edge-роутеры.
4. Создать Datastore Cluster из датастора(ов), на котором будут размещаться пользовательские edge-роутеры и служебные сервисы.
5. Создать Datastore Cluster из датастора(ов), на котором будут размещаться диски пользователей (можно использовать из пункта 4).
6. Создать dvSwitch, под которым будут создаваться пользовательские сети (порт-группы).

7.1. Создание management-сети

Создаем management-сеть, в которой развёрнута и работает ВМ с РУСТЭК-ЕСУ – ESU-box (настройки сети должны совпадать с настройками маршрутизируемой сети внутри РУСТЭК), она же портгруппа на dvSwitch в vSphere (требуется один VLAN). Необходимо учитывать, что в эту сеть будут подключены пользовательские роутеры для сегмента VMware (в разделе «*Настройка сети для роутеров (edge) сегмента VMware vSphere*» этого документа описана процедура, позволяющая изменить такое поведение, создав отдельную сеть для роутеров).

Таким образом, размер подсети напрямую влияет на максимальное число ВЦОДов. ВМ ESU-box с РУСТЭК-ЕСУ станет DHCP-сервером в этой подсети (также возможна установка в сеть, где уже присутствует DHCP сервер). В данном примере сеть называется vlan3058 (*Создание:* ***Рисунок 47****,* ***Рисунок 48****, Редактирование:* ***Рисунок 49*** *и* ***Рисунок 50***).

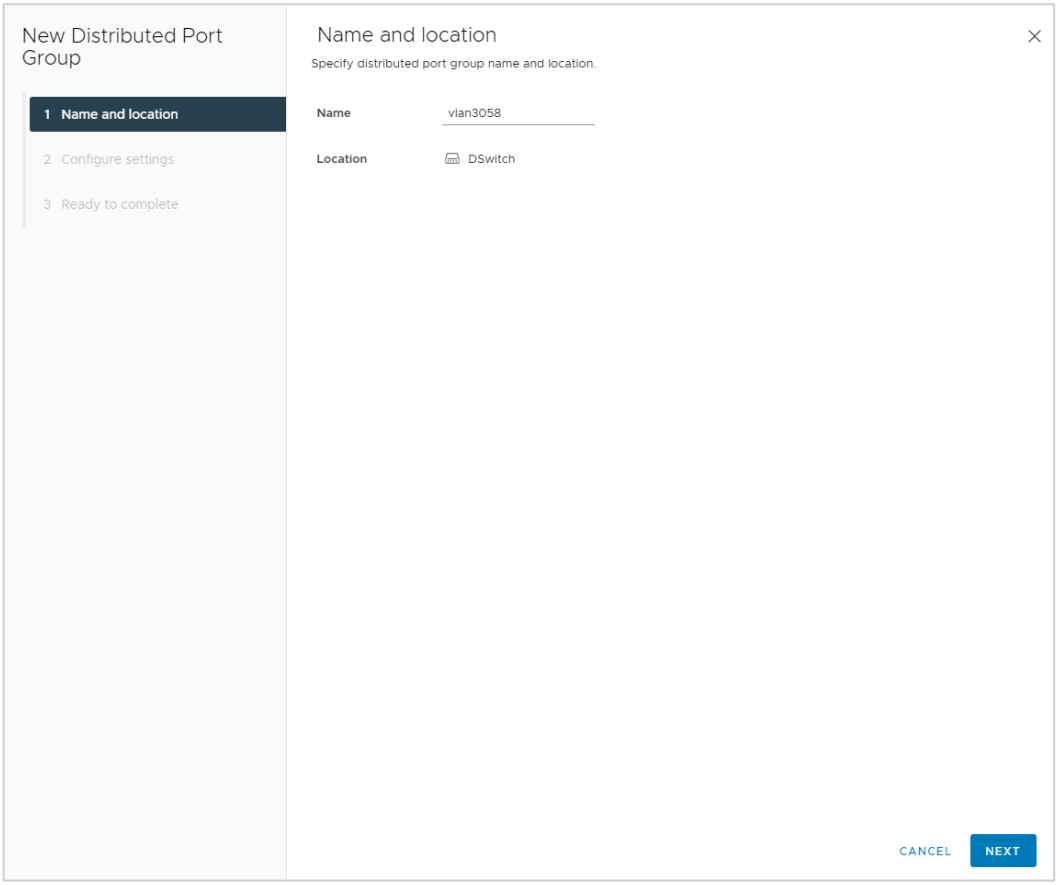

Рисунок 47

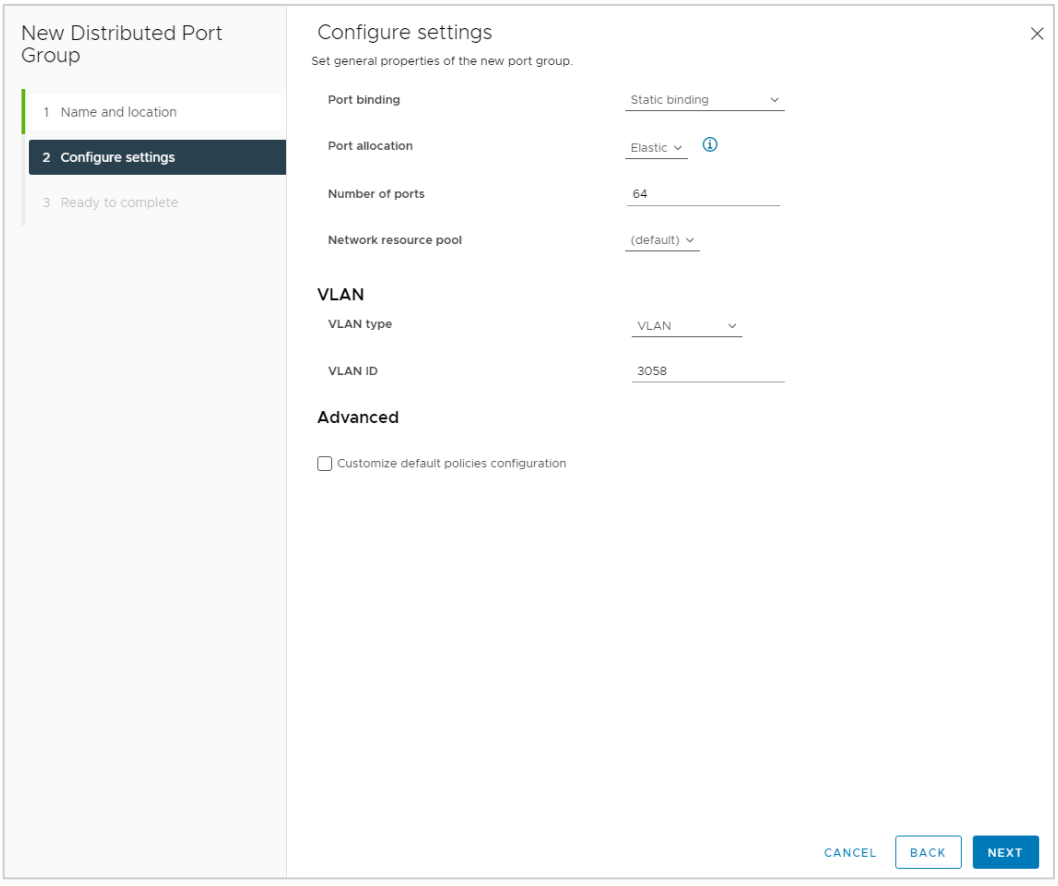

Рисунок 48

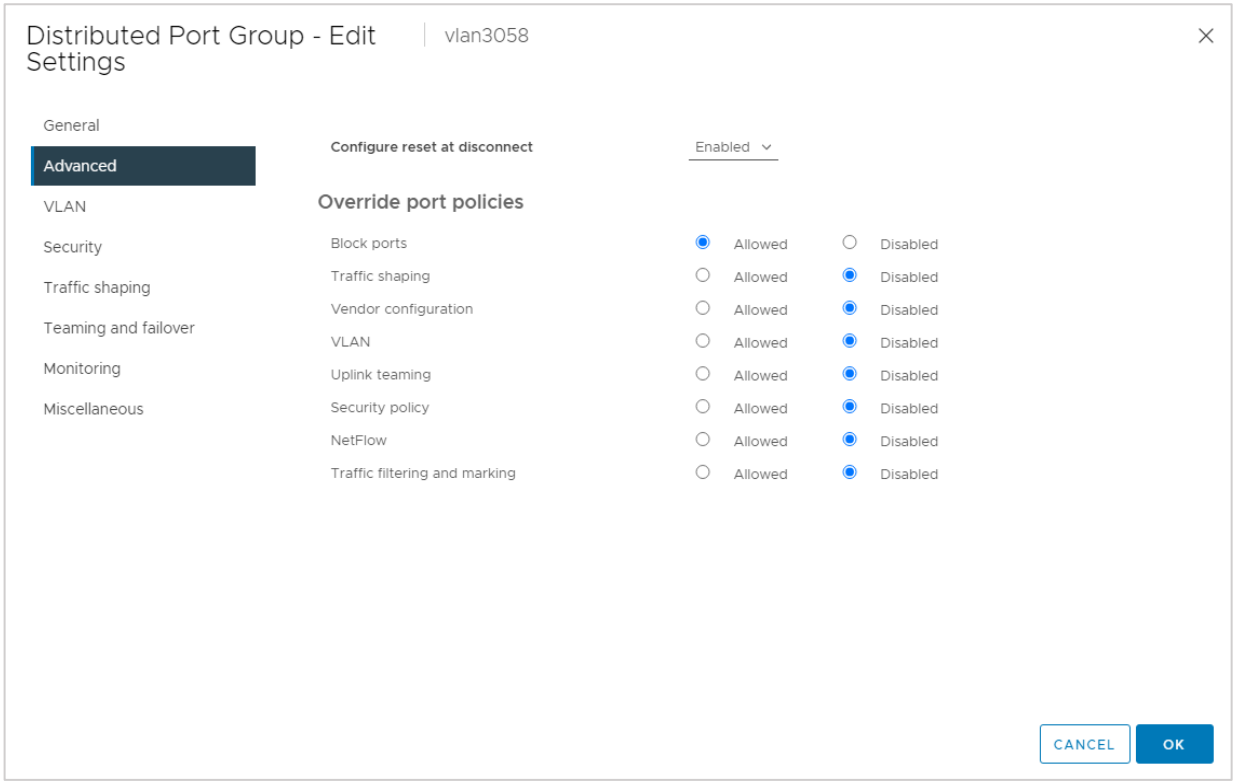

Рисунок 49

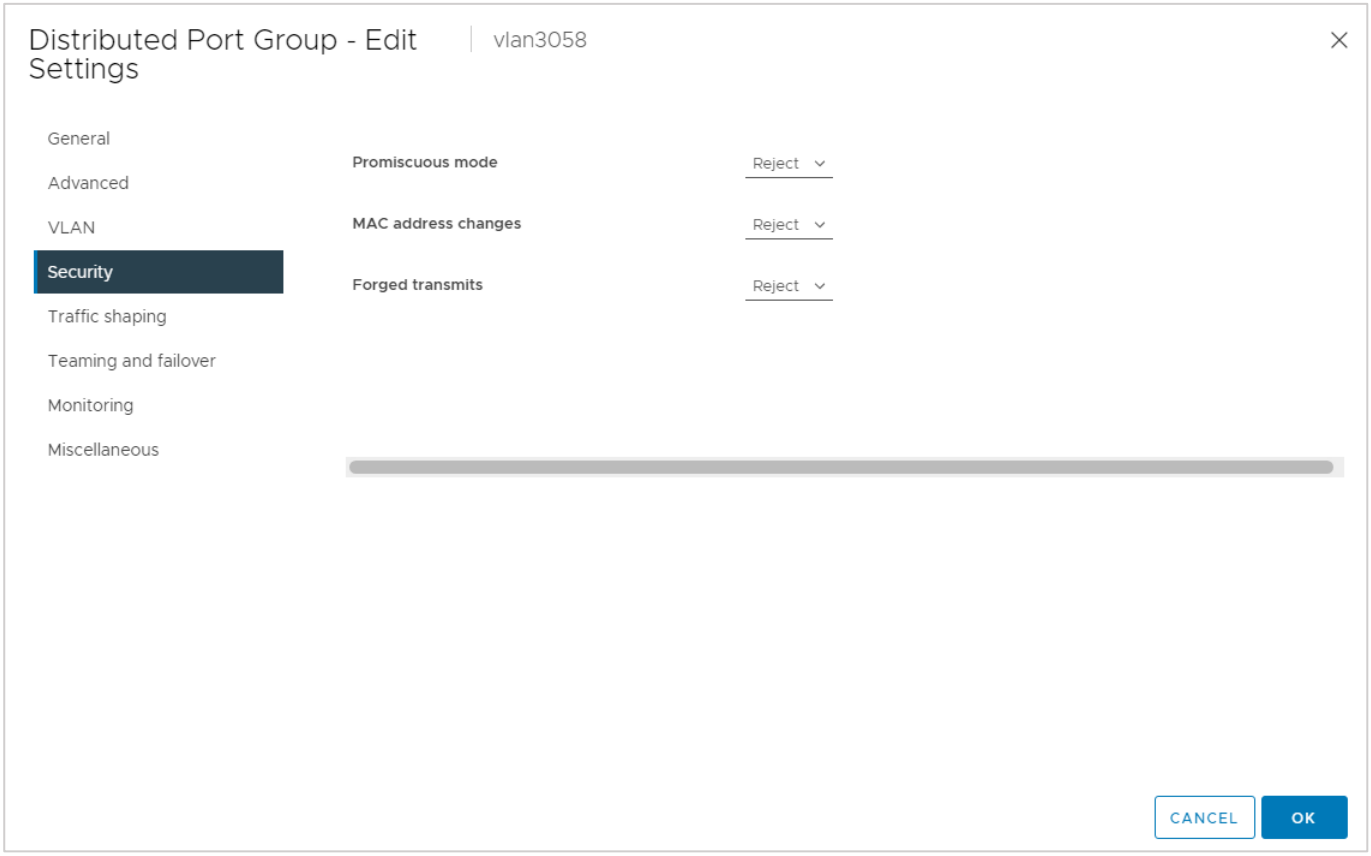

Рисунок 50

7.2. Создание директории для ВЦОДов клиентов

Создаем директорию, в которой будут располагаться ВЦОДы клиентов. Например, ESU3-Test, а в ней директорию Management (***Рисунок 51***)***:***

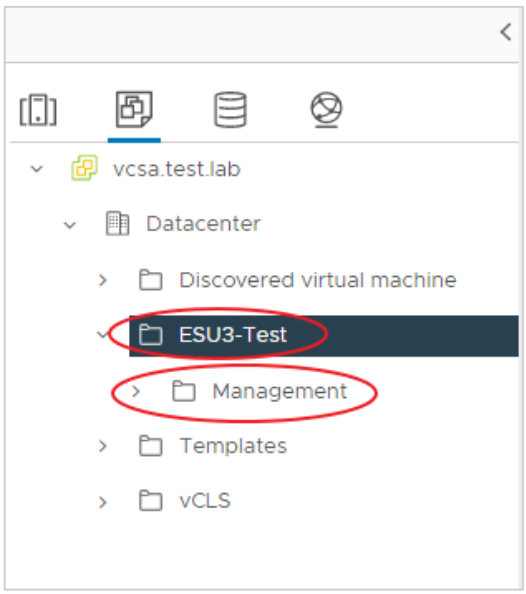

Рисунок 51

7.3. Настройка сетевых зон для сегмента VMware vSphere

Для создания сетевой зоны в панели управления РУСТЭК-ЕСУ переходим в меню **Инсталляция** – **Ресурсы** – **Сетевые зоны**, нажимаем «Создать сетевую зону» (***Рисунок 52***).

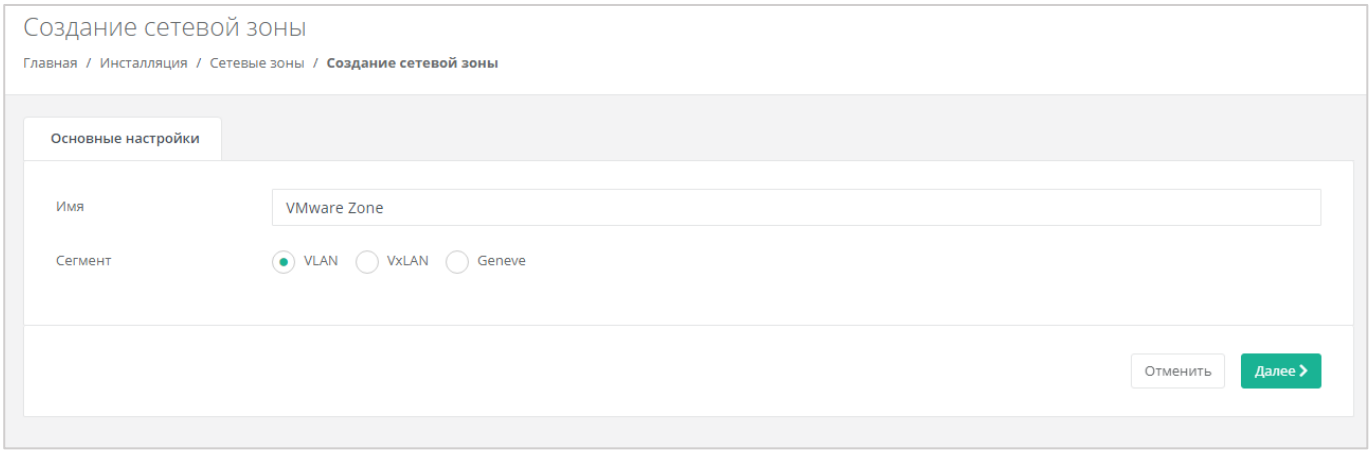

Рисунок 52

После этого нажимаем «Добавить пул» и добавляем необходимые пулы VLAN (***Рисунок 53***).

Диапазон 3060 - 3090 будем использовать для локальных пользовательских IP-адресов.

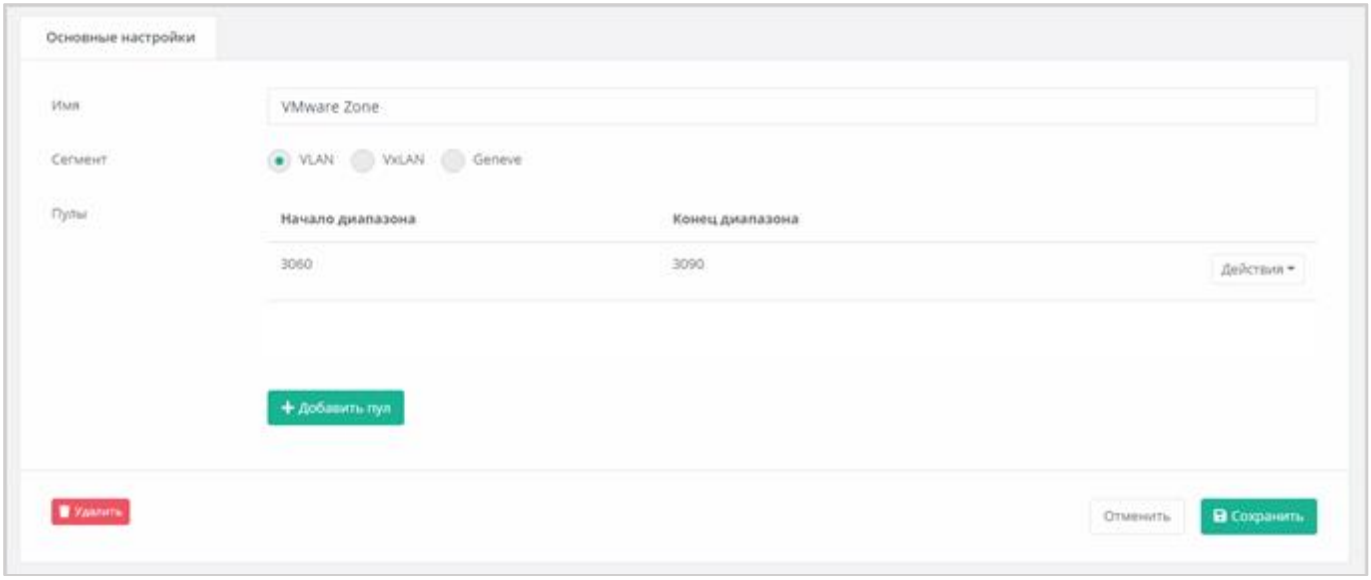

Рисунок 53

Аналогично создаём вторую сетевую зону для внешней сети (***Рисунок 54***).

VLAN 3227 будем использовать для публичных IP-адресов пользовательских ВЦОДов – устанавливаем его в начало и конец диапазона.

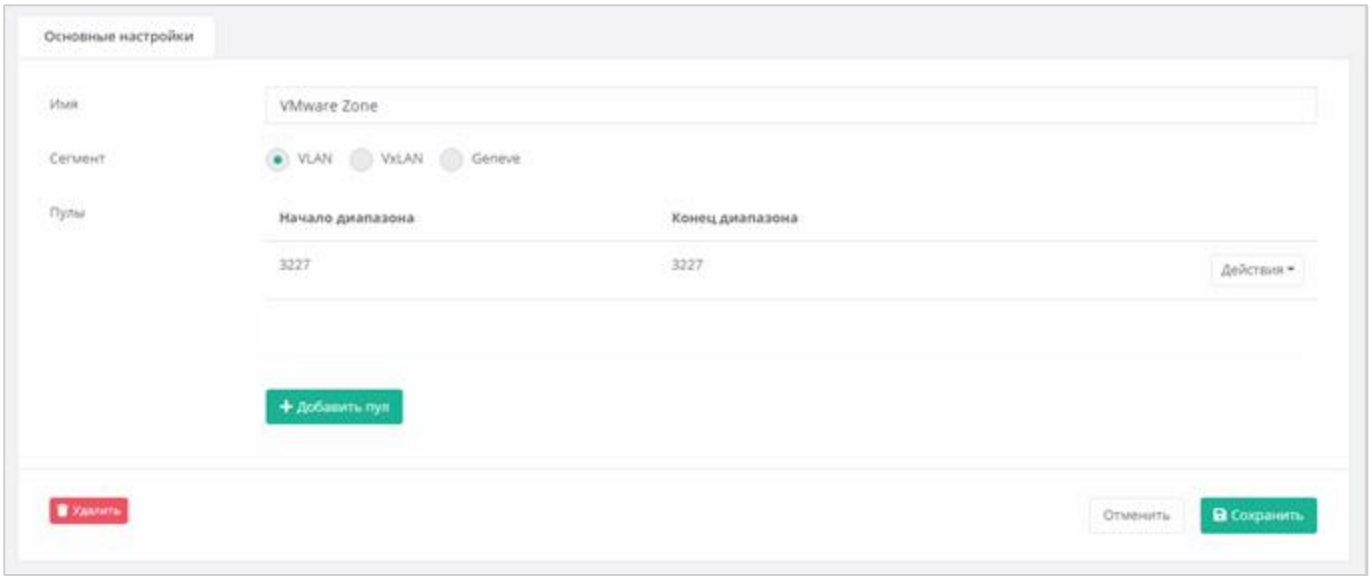

Рисунок 54

Заводим внешнюю сеть для сегмента VMware vSphere.

Переходим в **Инсталляция** – **Ресурсы** – **Сети и IP**. Нажимаем «Создать Сеть» (***Рисунок 55***).

- Название – любое.
- Сетевая зона – созданная ранее для внешней сети VMware-сегмента.
- VID/VNID – VLAN внешней сети (в нашем случае 3227).

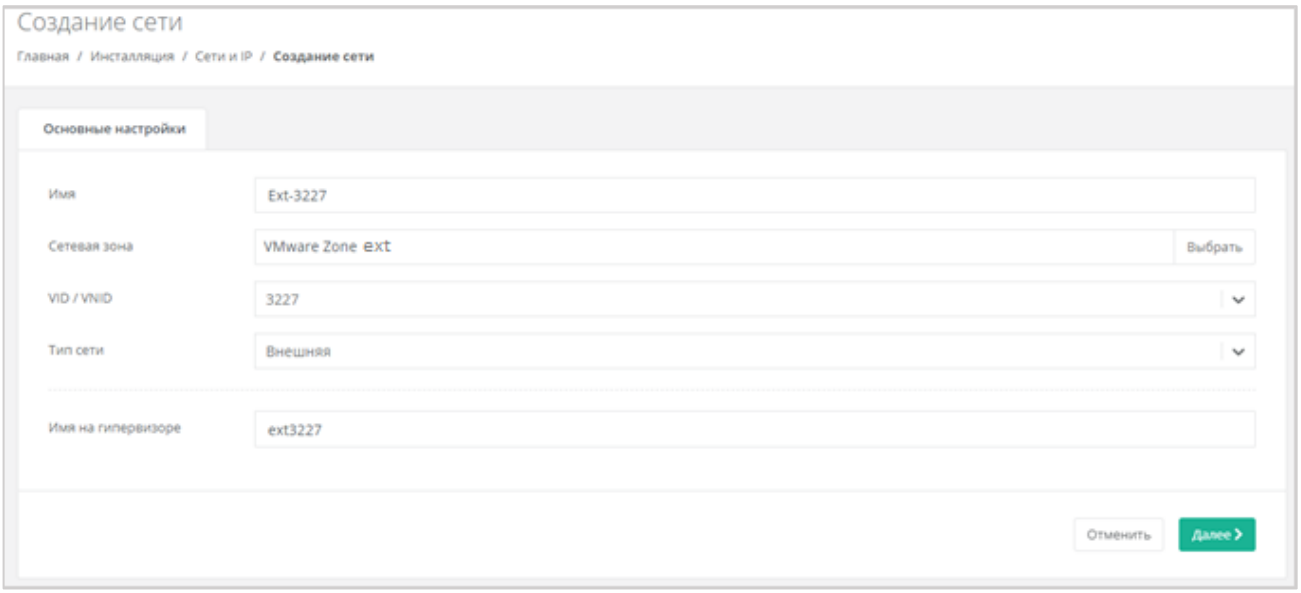

Рисунок 55

Затем нажимаем на кнопку «Добавить подсеть».

DHCP должен быть **выключен**, CIDR необходимо указывать полный. Если нужно порезать диапазон выдаваемых IP, можно указать произвольный диапазон (***Рисунок 56***).

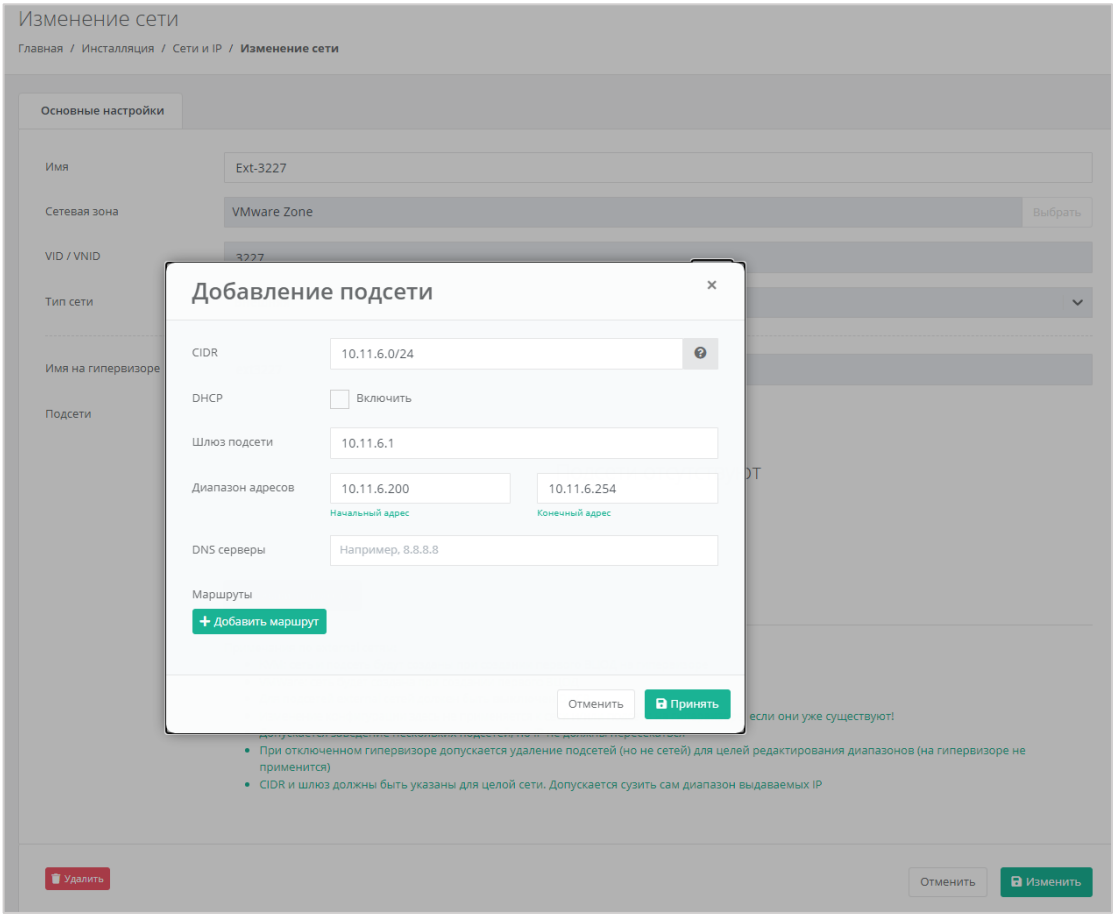

Рисунок 56

7.4. Настройка vSphere-раннера РУСТЭК-ЕСУ

В настройках веб-интерфейса переходим в **Инсталляция** – **Система** – **Раннеры** и конфигурируем vSphere-раннер РУСТЭК-ЕСУ (***Рисунок 57***). Указываем:

- IP-адрес сервера vCenter;
- имя пользователя и пароль для доступа к vCenter (учётная запись должна быть с правами администратора);
- название дата-центра – название должно соответствовать фактическому названию в vSphere (например, Datacenter, см. ***Рисунок 58***);
- название DVSwitch, на котором будут создаваться пользовательские сети (порт-группы).

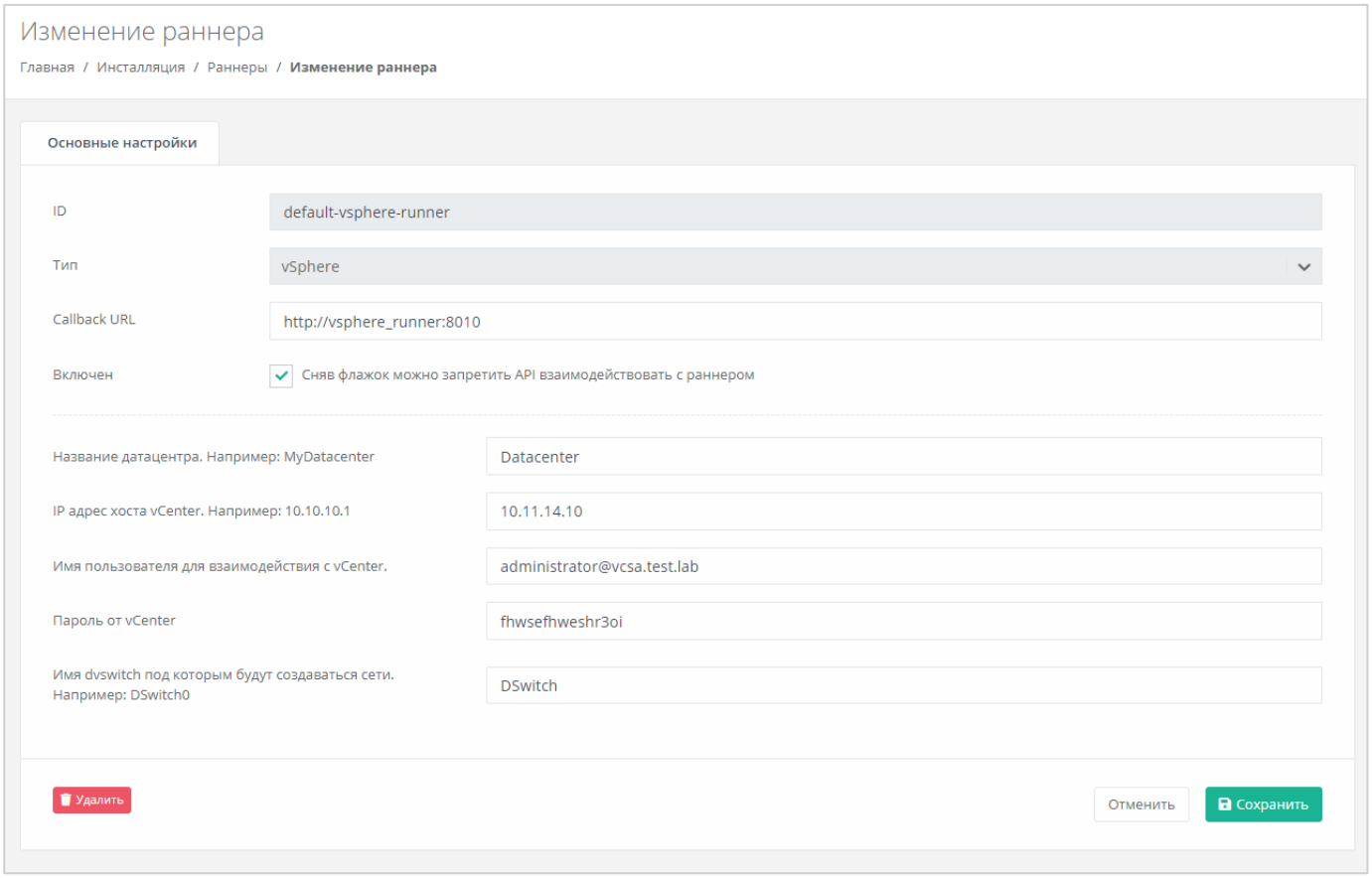

Рисунок 57

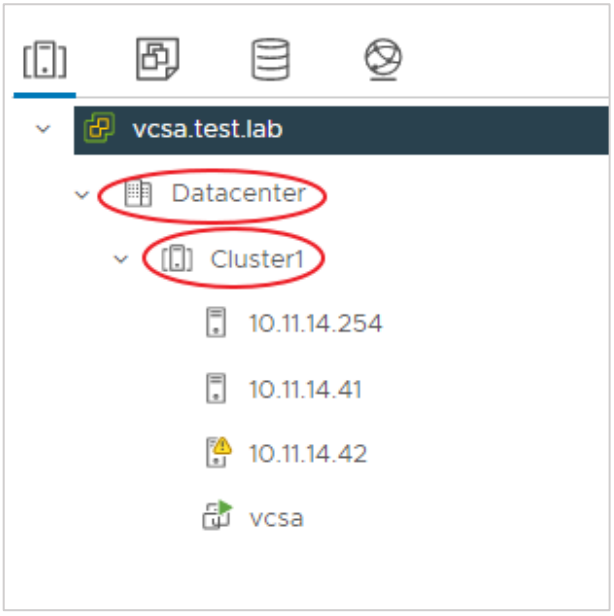

Рисунок 58

Если настройки введены правильно, индикатор vSphere-раннера должен быть зелёным (***Рисунок 59***).

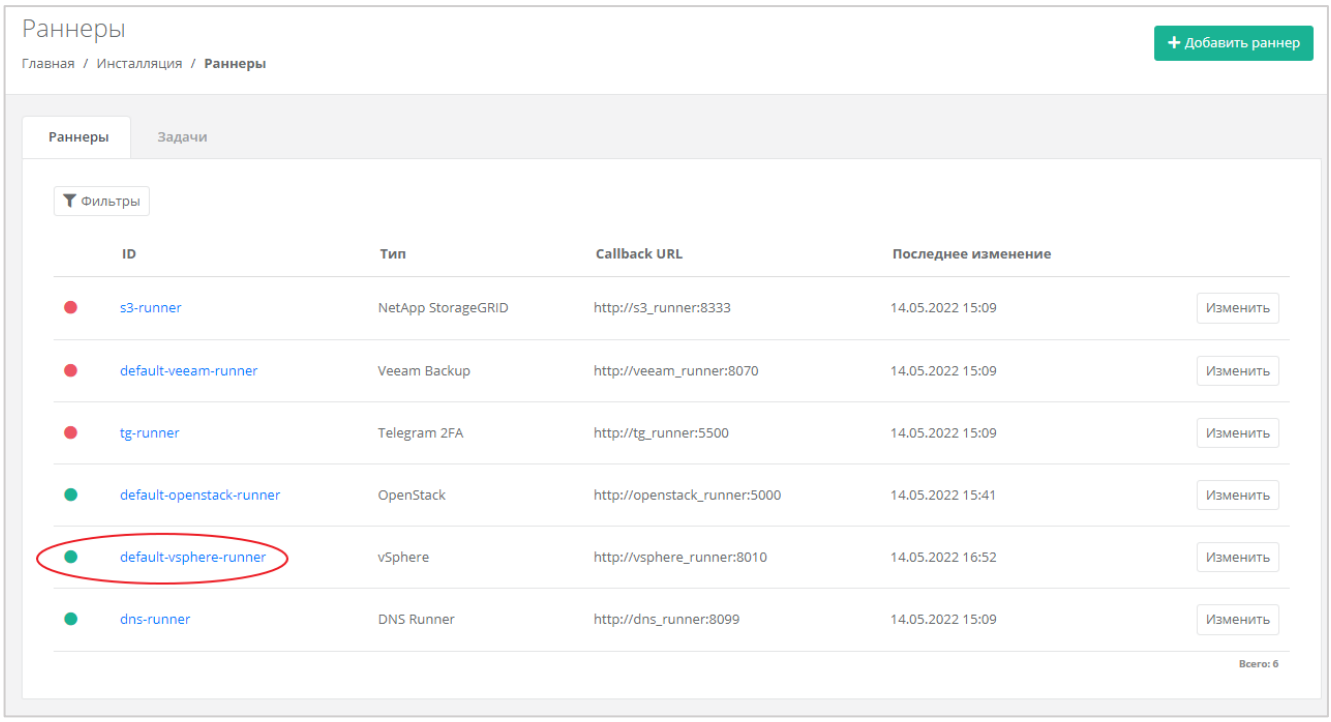

Рисунок 59

Создадим новый токен для пользователя runner – он понадобится для дальнейших настроек (***Рисунок 60***).

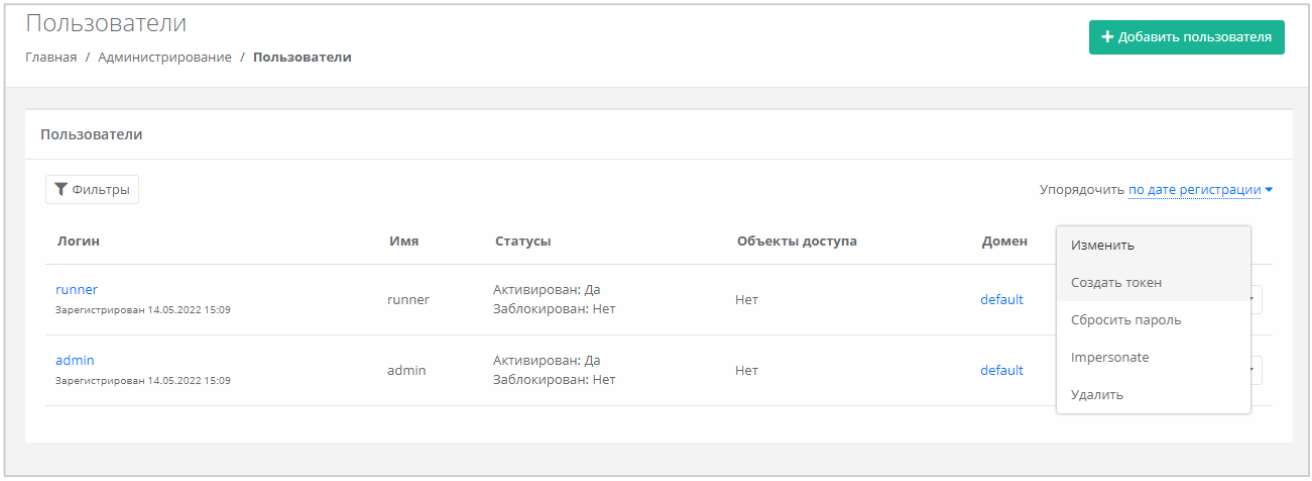

Рисунок 60

7.5. Настройка ресурсного пула для сегмента VMware vSphere

Далее необходимо изменить настройки ресурсного пула (***Рисунок 61***).

Переходим в **Инсталляция** – **Ресурсы** – **Ресурсные пулы**.

На вкладке ***Основные настройки*** выбираем VMware Hypervisor.

Указываем:

- Тип – VMware.
- Раннер (в нашем случае – *default-vsphere-runner*).
- Сетевую зону (в нашем случае – vSphere Zone).
- Внешнюю сеть (в нашем случае ext-3227).
- Устанавливаем чек-бокс «Включен».

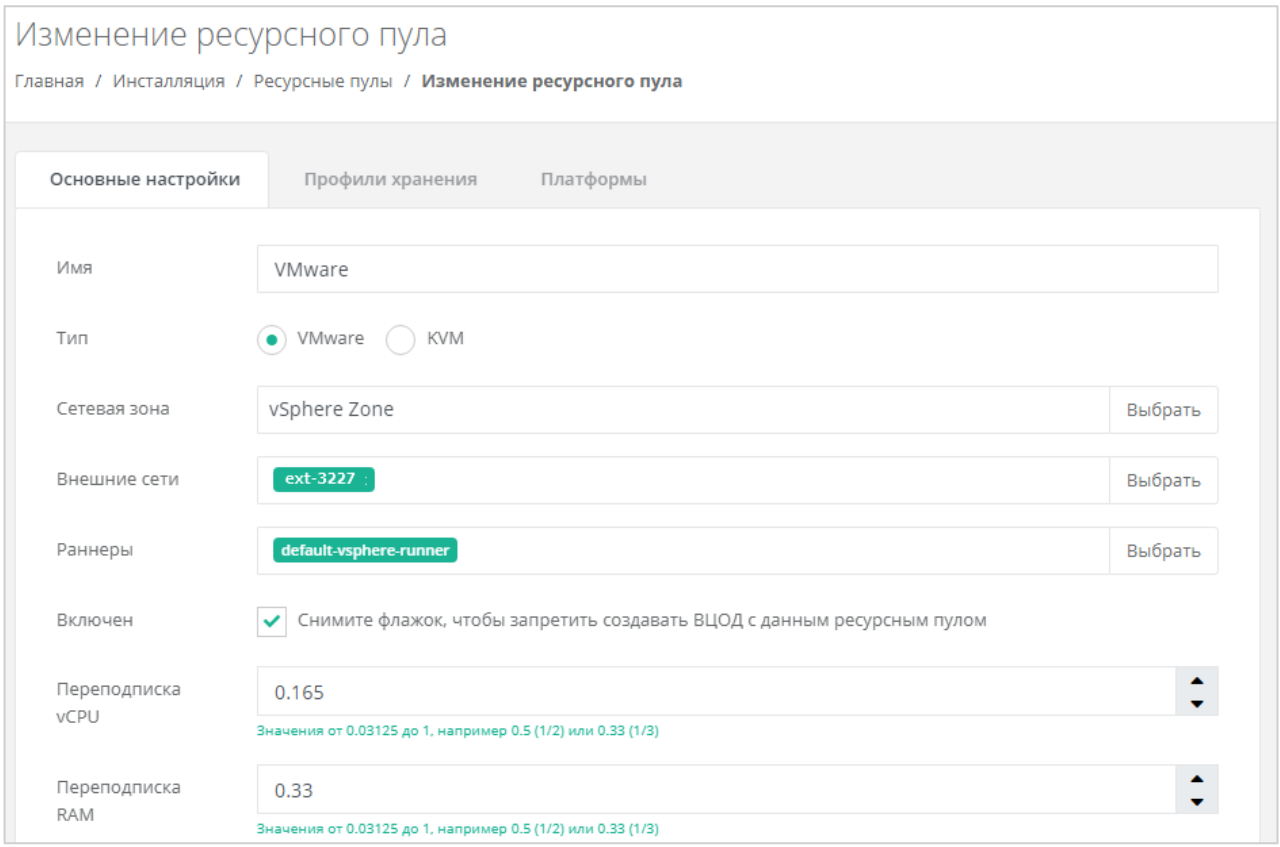

Рисунок 61

Далее необходимо настроить переподписку vCPU и RAM (***Рисунок 62***).

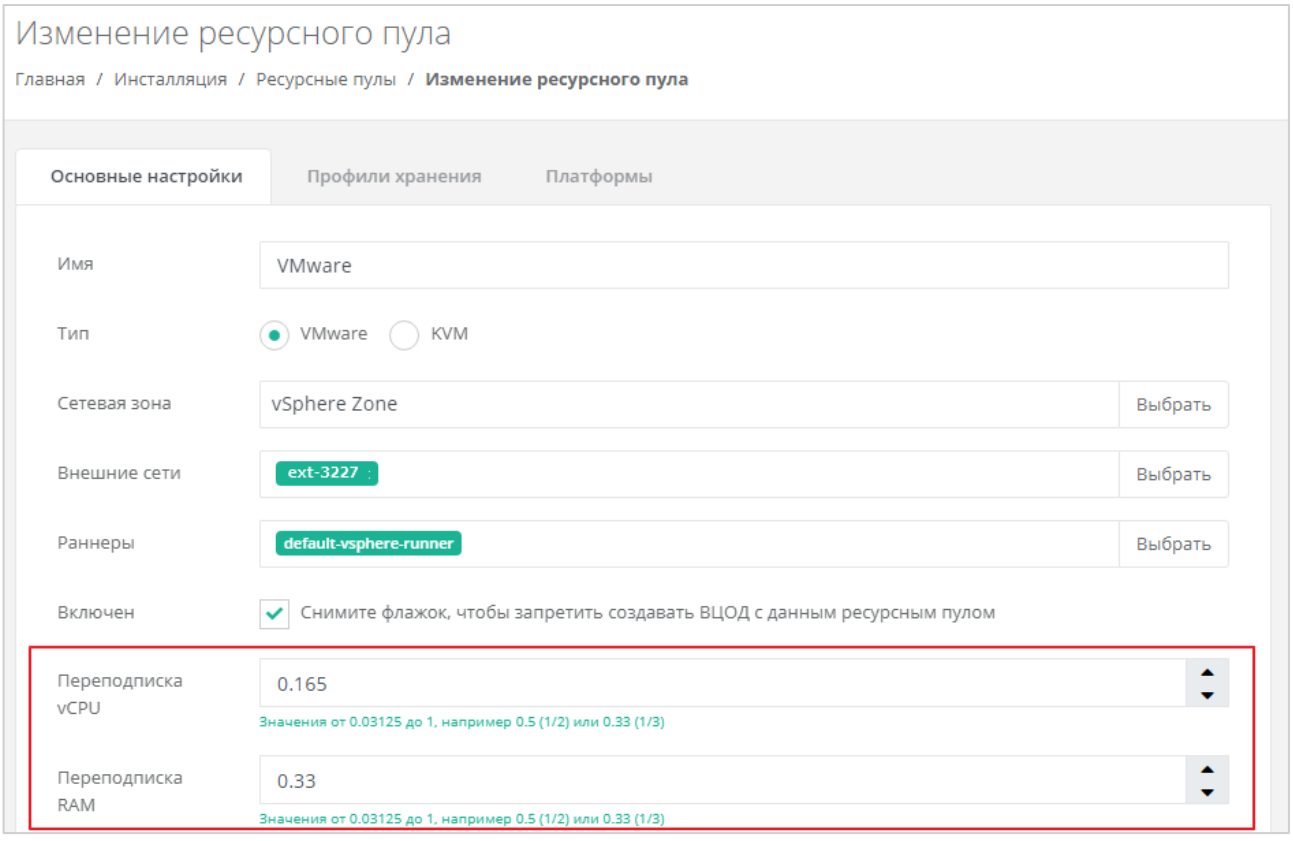

Рисунок 62

После этого необходимо настроить ограничения по vCPU, RAM и дискам на один сервер (***Рисунок 63***).

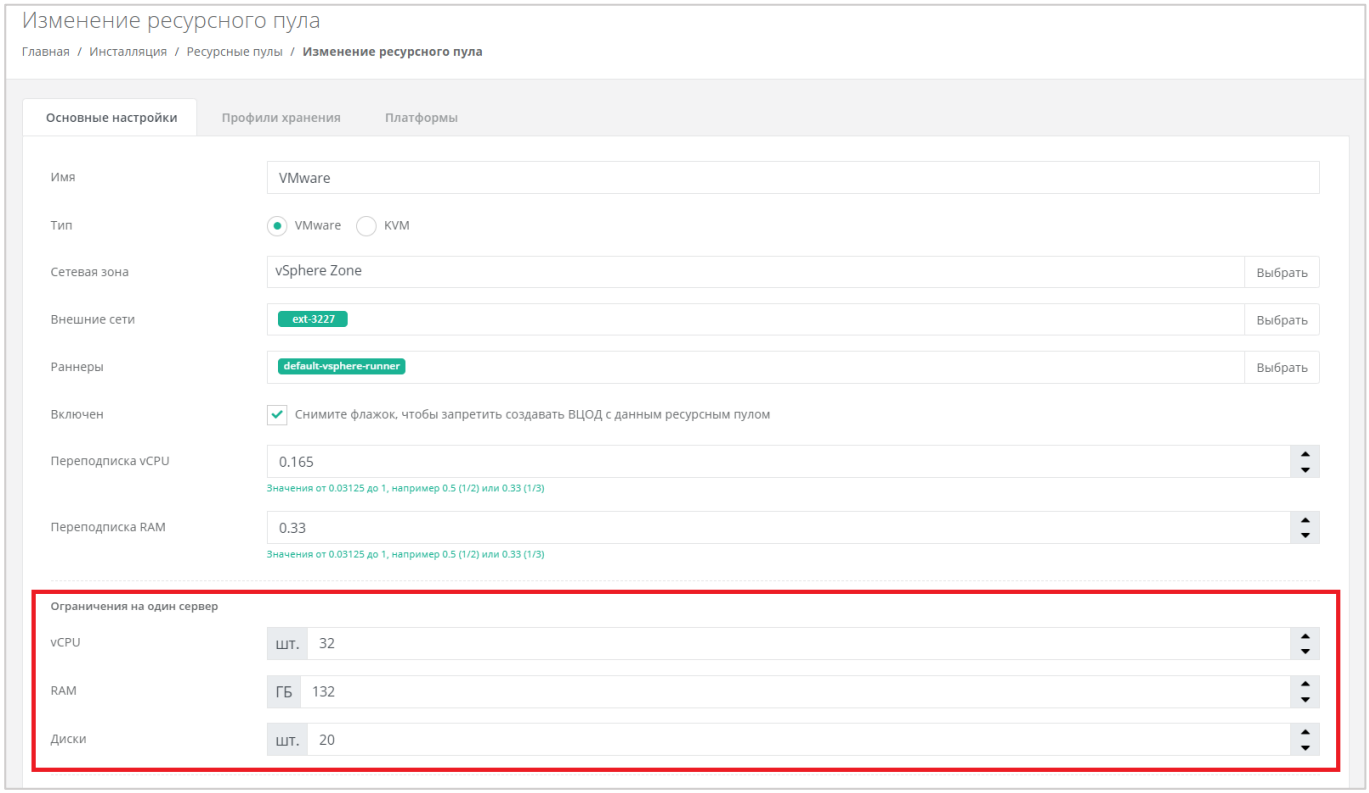

Рисунок 63

Ниже на той же станице указываем следующие настройки (***Рисунок 64***):

- название шаблона роутера – укажем «edge»;
- название management-сети (порт-группы), в которой работает РУСТЭК-ЕСУ;
- название служебного датастора, на котором будут размещаться пользовательские роутеры и служебные сервисы;
- адрес РУСТЭК-ЕСУ в management-сети, по которому будет доступно API;
- токен, который будет использоваться Edge-роутерами для работы с РУСТЭК-ЕСУ (был создан шагом выше).

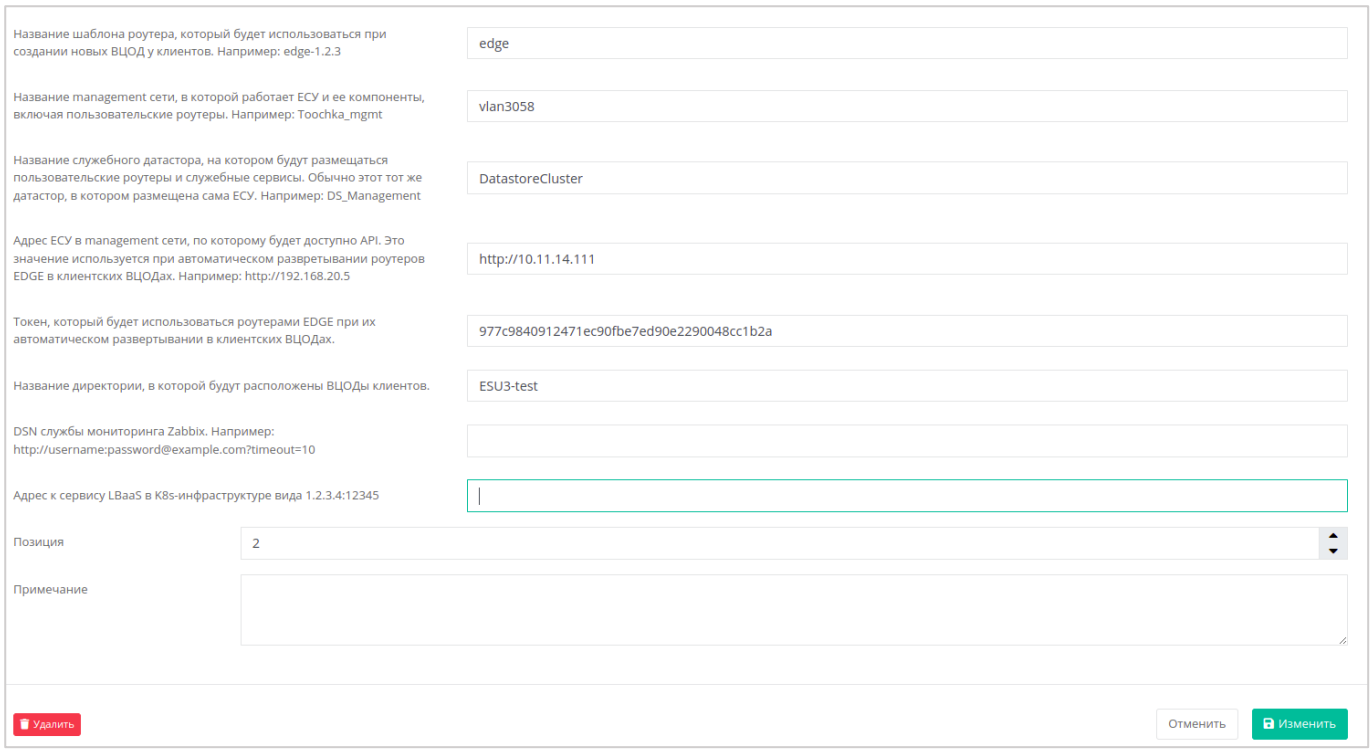

Рисунок 64

На вкладке ***Профили хранения*** добавляем профили хранения (***Рисунок 65***) – указываем имя, отображаемое название, название Storage DRS-кластера vSphere, который будет использоваться для хранения дисков ВМ и выбираем биллинг-класс (***Рисунок 66***).

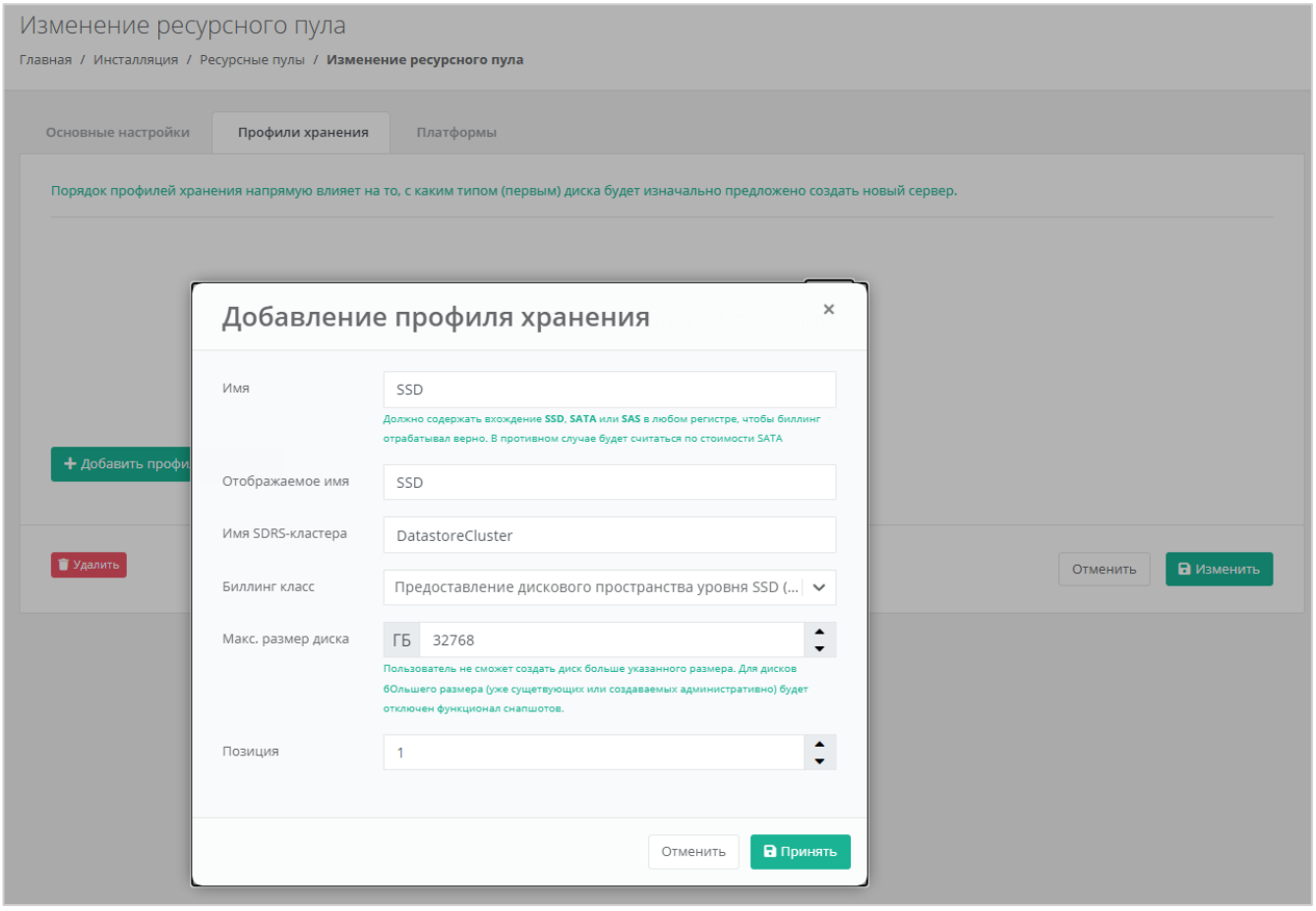

Рисунок 65

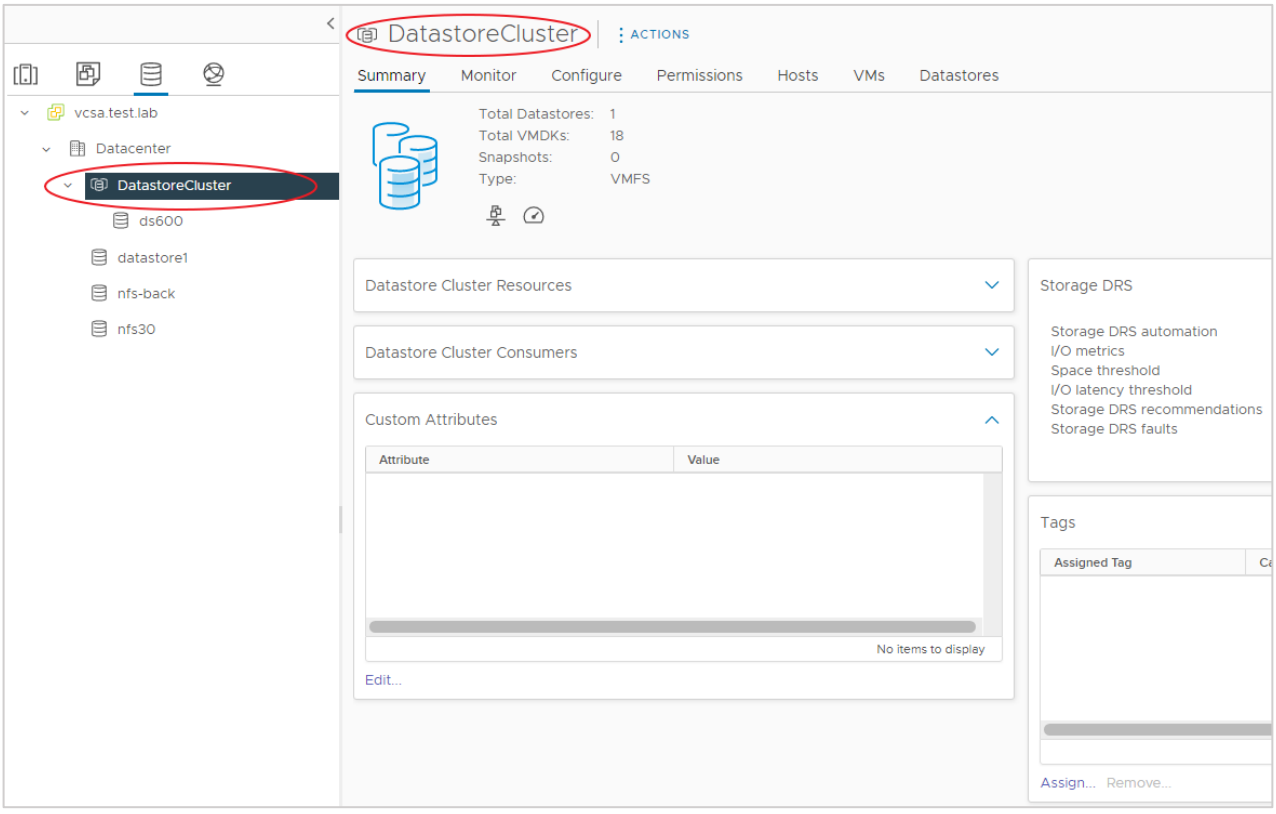

Рисунок 66

Далее переходим на вкладку ***Платформы*** (***Рисунок 61***) и нажимаем кнопку «Добавить платформу», в открывшемся окне указываем имя созданного кластера (***в нашем примере – Cluster1*** (***Рисунок 67***).

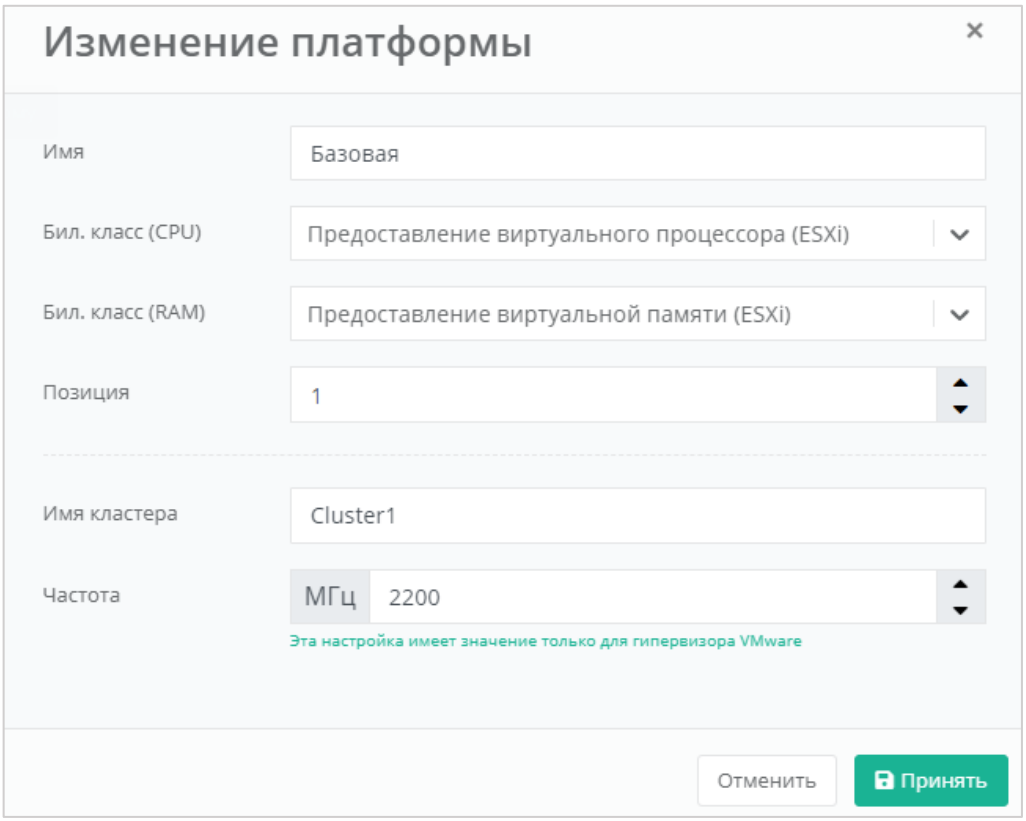

Рисунок 67

После того, как введены все настройки, в форме изменения ресурсного пула нажимаем кнопку «Принять».

7.6. Развёртывание Edge-роутера

Теперь, когда vSphere runner и ресурсный пул настроены, необходимо произвести развёртывание Edge-роутера. Развёртывание будет произведено на все ресурсные пулы VMware, настроенные в системе.

Для развёртывания необходимо зайти по SSH на ESU-box (стандартная УЗ deploy:1-qpALzm/), посмотреть, какая последняя версия роутера доступна в данной версии (ls -lah | grep edge*.ova) и выполнить команду:

toochkactl edge-deploy --filename edge-x.x.x

где x.x.x — последняя доступная версия.

!!!Важно!!! В целях безопасности настоятельно рекомендуем изменить логин и пароль учётной записи после настройки.

Инструмент toochkactl произведёт заливку и развертывание шаблона роутера (в формате .ova) на ресурсных пулах (***Рисунок 68***).

```
deploy@localhost:~$ toochkactl edge-deploy --filename edge-1.2.7.ova
 _____                 _     _
(_   _)               ( )   ( )
  | |   _    _ _    ___ | |__ | |/')    _ _
  | | /'_`\ /'_`\ /'___)|  _ `\| , <   /'_` )
  | |( (_) )( (_) )( (___ | | | || |\`\ ( (_| |
  (_)`\___/'`\___/'`\____)(_) (_)(_) (_)`\__,_)


Config file: /opt/box/toochka.conf
Upload EDGE template...
sudo: unable to resolve host localhost: Name or service not known
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [localhost] ***************************************************************************************************************************************
*********************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
*********************************************************************
ok: [localhost]

TASK [deploy_edge : Deploy EDGE] ***********************************************************************************************************************
*********************************************************************
[WARNING]: Skipping plugin (/usr/local/lib/python3.7/dist-packages/ansible/plugins/filter/core.py) as it seems to be invalid: cannot
 import name 'environmentfilter' from 'jinja2.filters'
(/usr/local/lib/python3.7/dist-packages/jinja2/filters.py)
[WARNING]: Skipping plugin (/usr/local/lib/python3.7/dist-packages/ansible/plugins/filter/mathstuff.py) as it seems to be invalid: c
annot import name 'environmentfilter' from 'jinja2.filters'
(/usr/local/lib/python3.7/dist-packages/jinja2/filters.py)
changed: [localhost -> localhost]

PLAY RECAP *********************************************************************************************************************************************
*********************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

deploy@localhost:~$
```

Рисунок 68

Обратите внимание, что силу особенностей развёртывания шаблона у вас должна быть стандартная портгруппа с названием «VM Network».

После завершения развёртывания на vSphere появится выключенная ВМ с названием «*edge-x.x.x*», а в настройках ресурсного пула вместо «edge» будет прописана актуальная версия роутера (***Рисунок 69***).

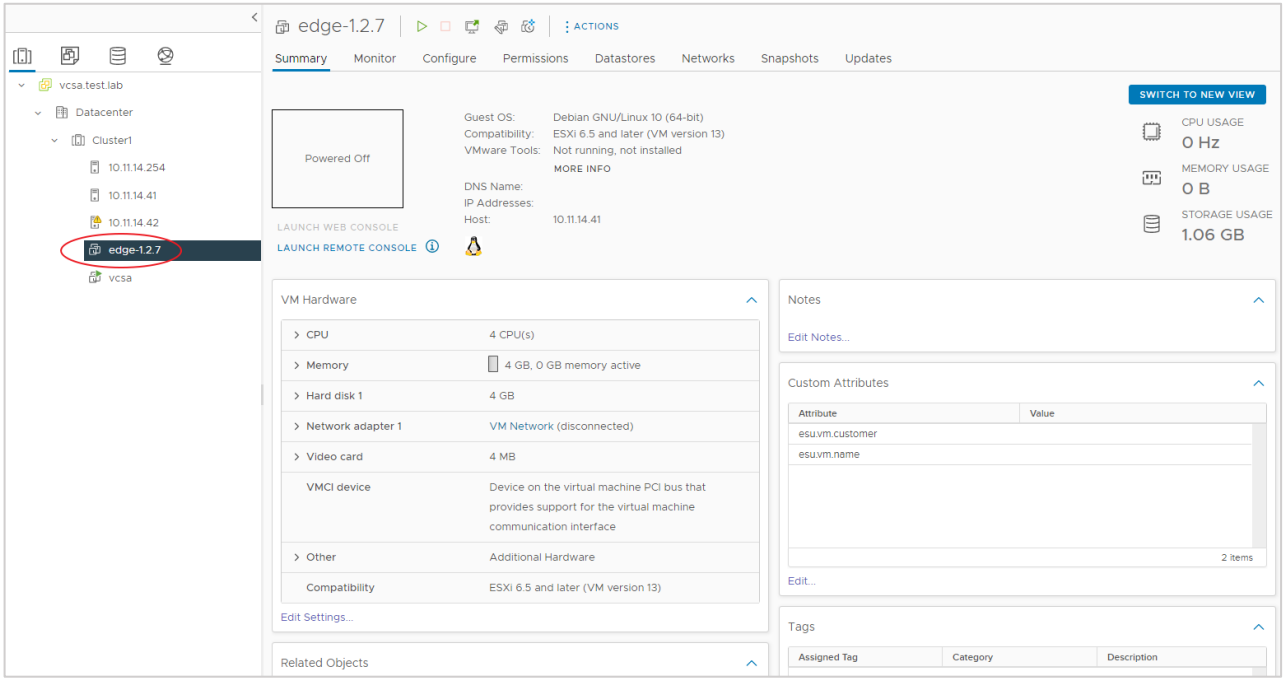

Рисунок 69

После этого Система будет готова к созданию ВЦОД в сегменте VMware.

8. Создание шаблонов ВМ для сегмента VMware vSphere

Для создания шаблона ВМ необходим образ ОС с cloud-init в формате .ova.

Далее будет рассмотрен пример создания шаблона ВМ с операционной системой Ubuntu 18.04 LTS.

Ссылка на используемый в примере образ:

https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.ova

Заходим в vSphere Client (***Рисунок 70***) и в целевом дата-центре производим развёртывание .ova-шаблона.

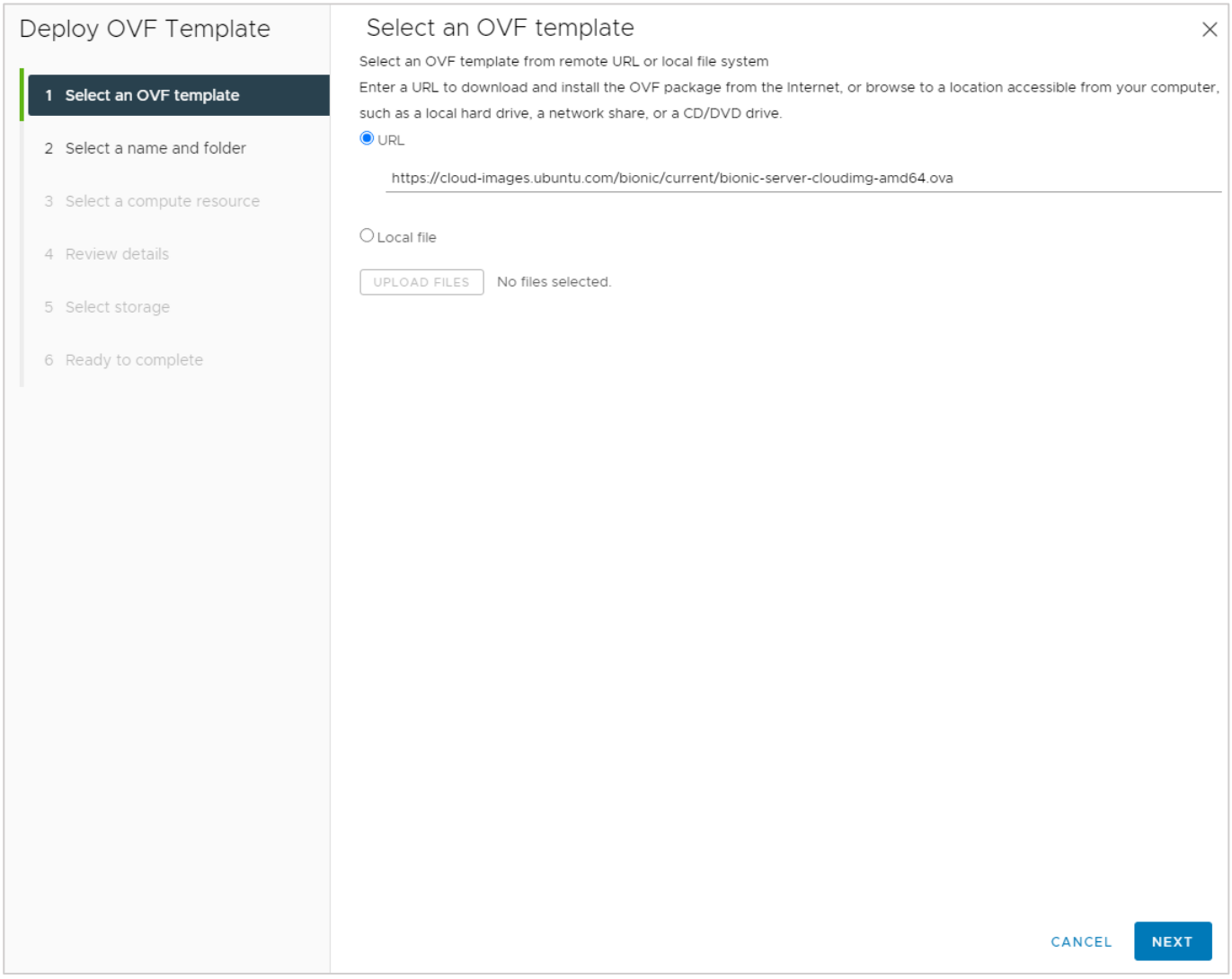

Рисунок 70

Далее в мастере деплоя (***Рисунок 71 – Рисунок 76***) необходимо указать имя создаваемой ВМ, кластер, на котором она будет развернута, хранилище, на котором будет лежать ВМ (при указании типа диска нужно указать «**Thin provision**»).

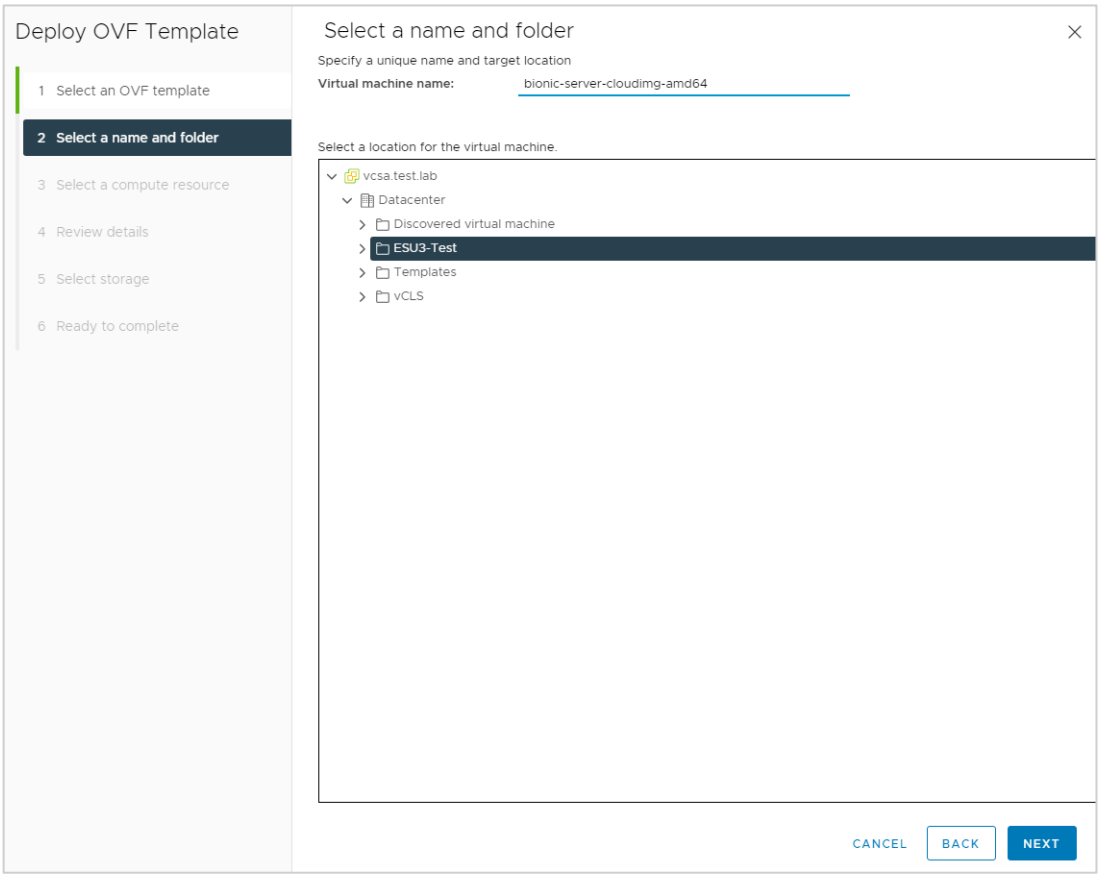

Рисунок 71

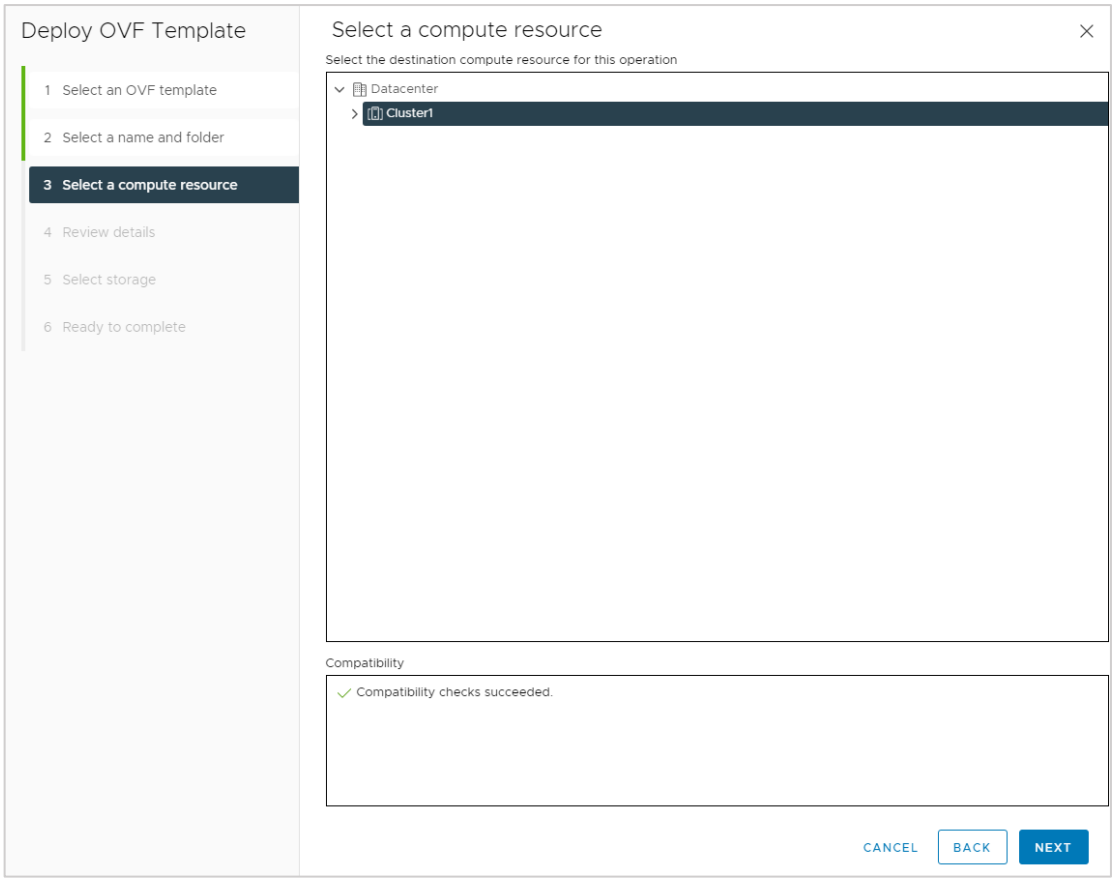

Рисунок 72

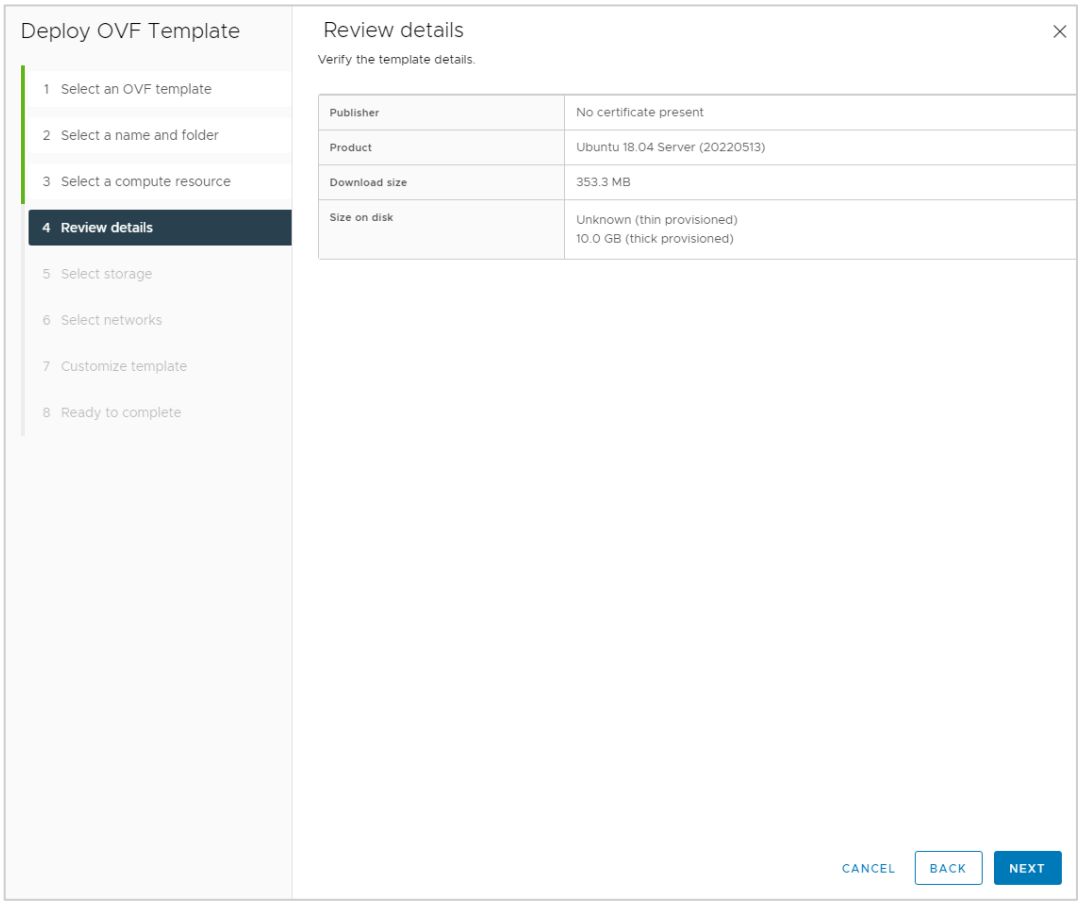

Рисунок 73

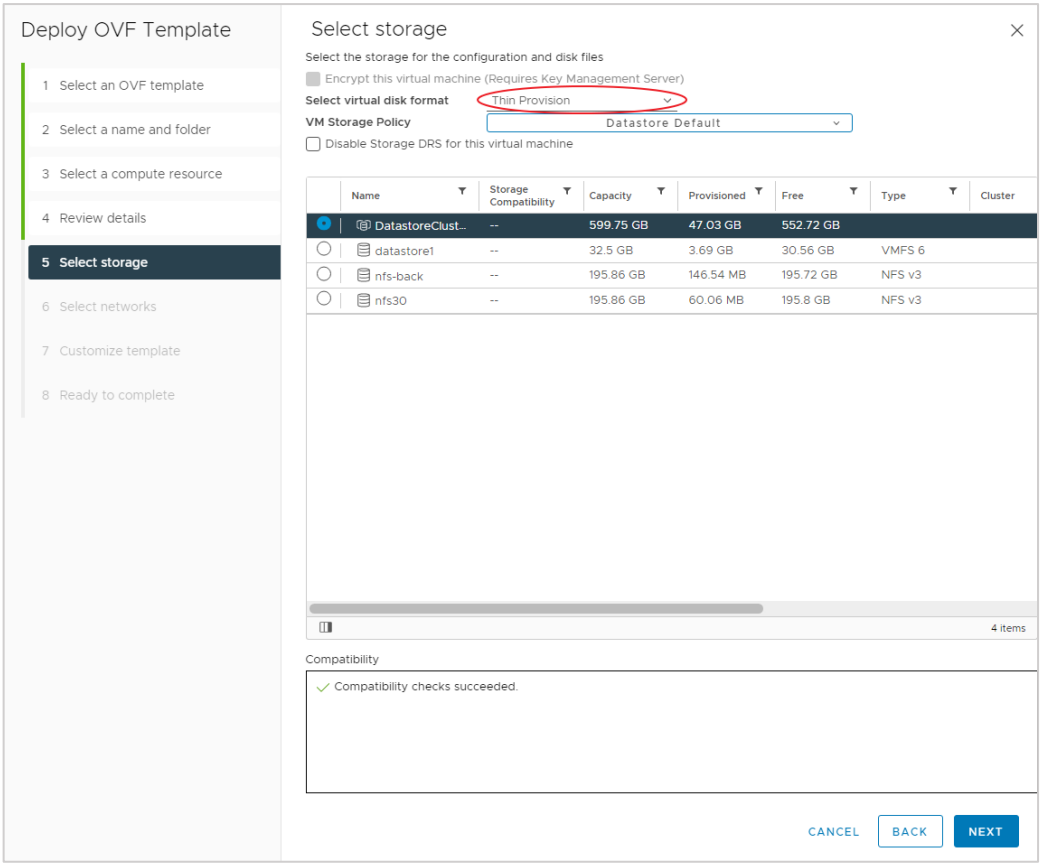

Рисунок 74

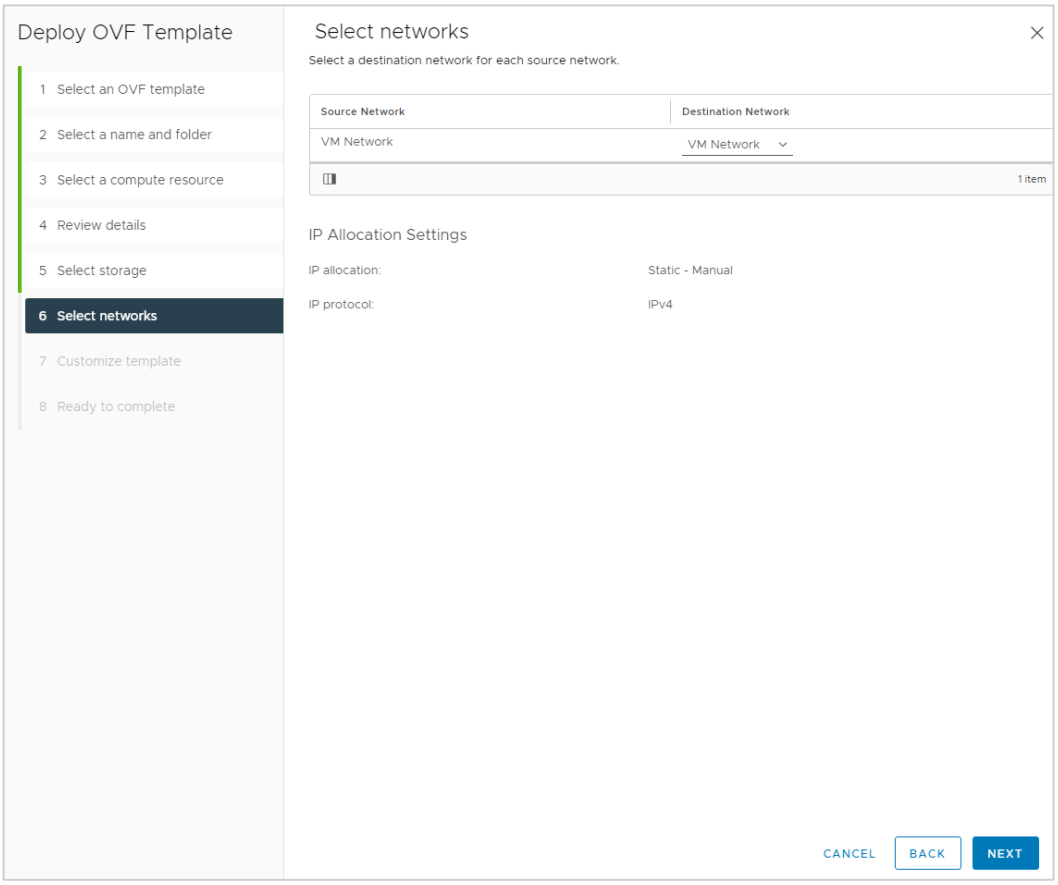

Рисунок 75

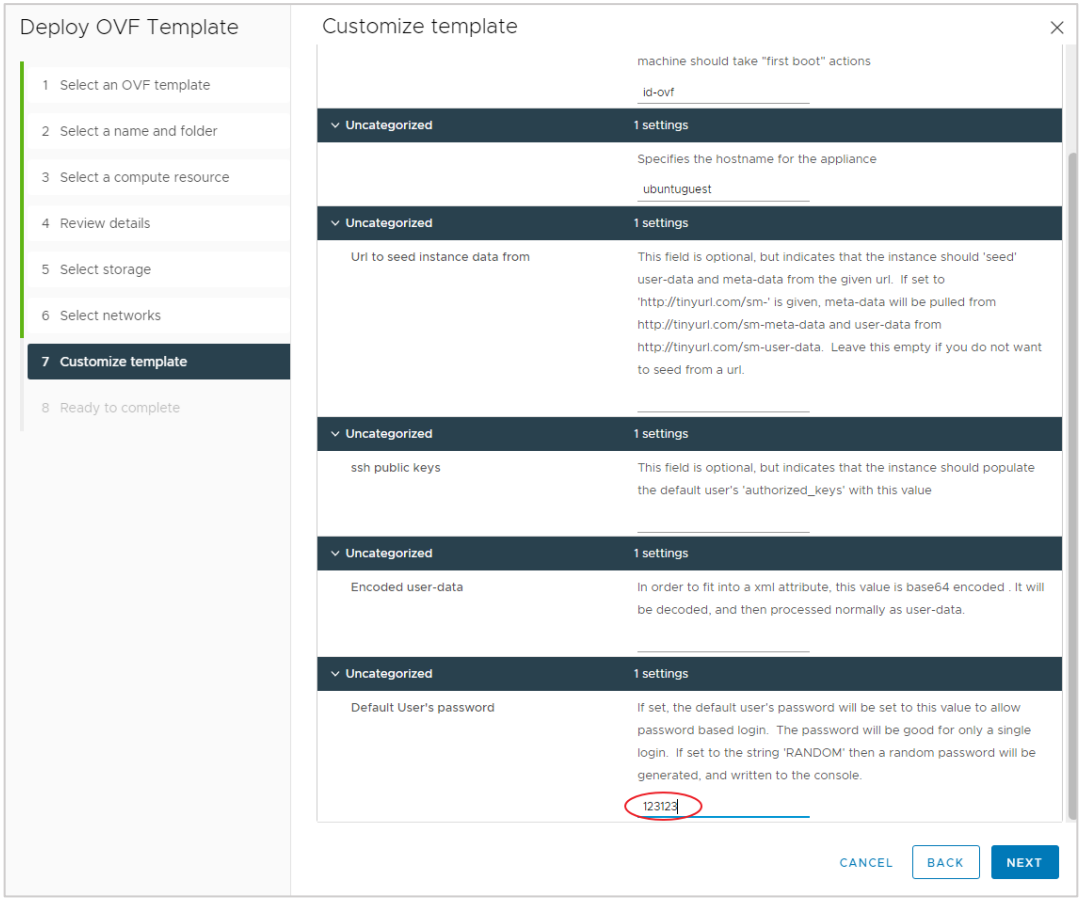

Рисунок 76

Дожидаемся завершения развёртывания .ova-шаблона. Ставим пароль на свое усмотрение (***Рисунок 77***).

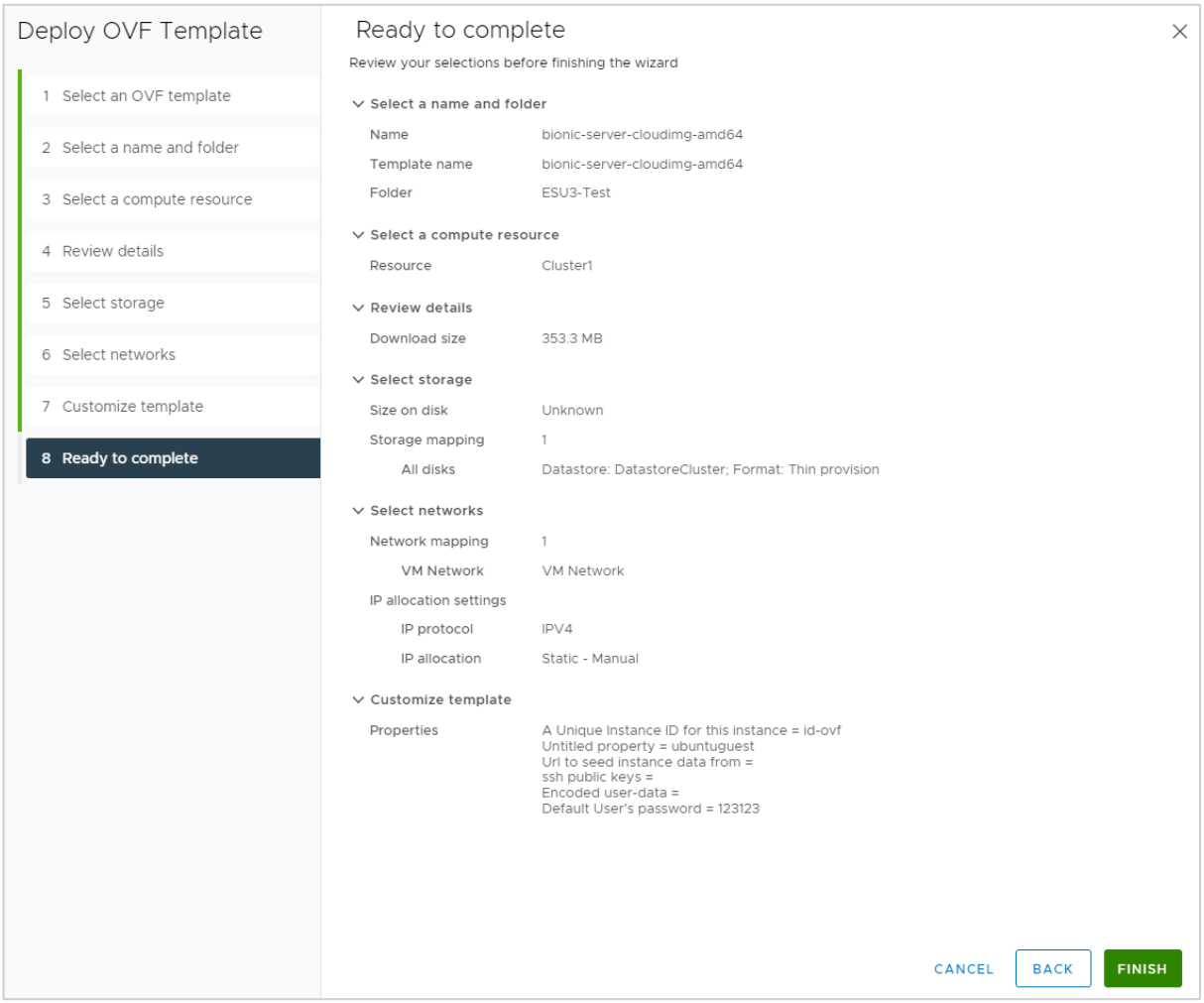

Рисунок 77

Далее необходимо отредактировать настройки ВМ (***Рисунок 78***) – выставляем необходимый нам тип SCSI-контроллера (VMware Paravirtual), удаляем сетевой адаптер, указываем для CD-ROM IDE 0:0 и проверяем у Hard Disk type Thin Provision.

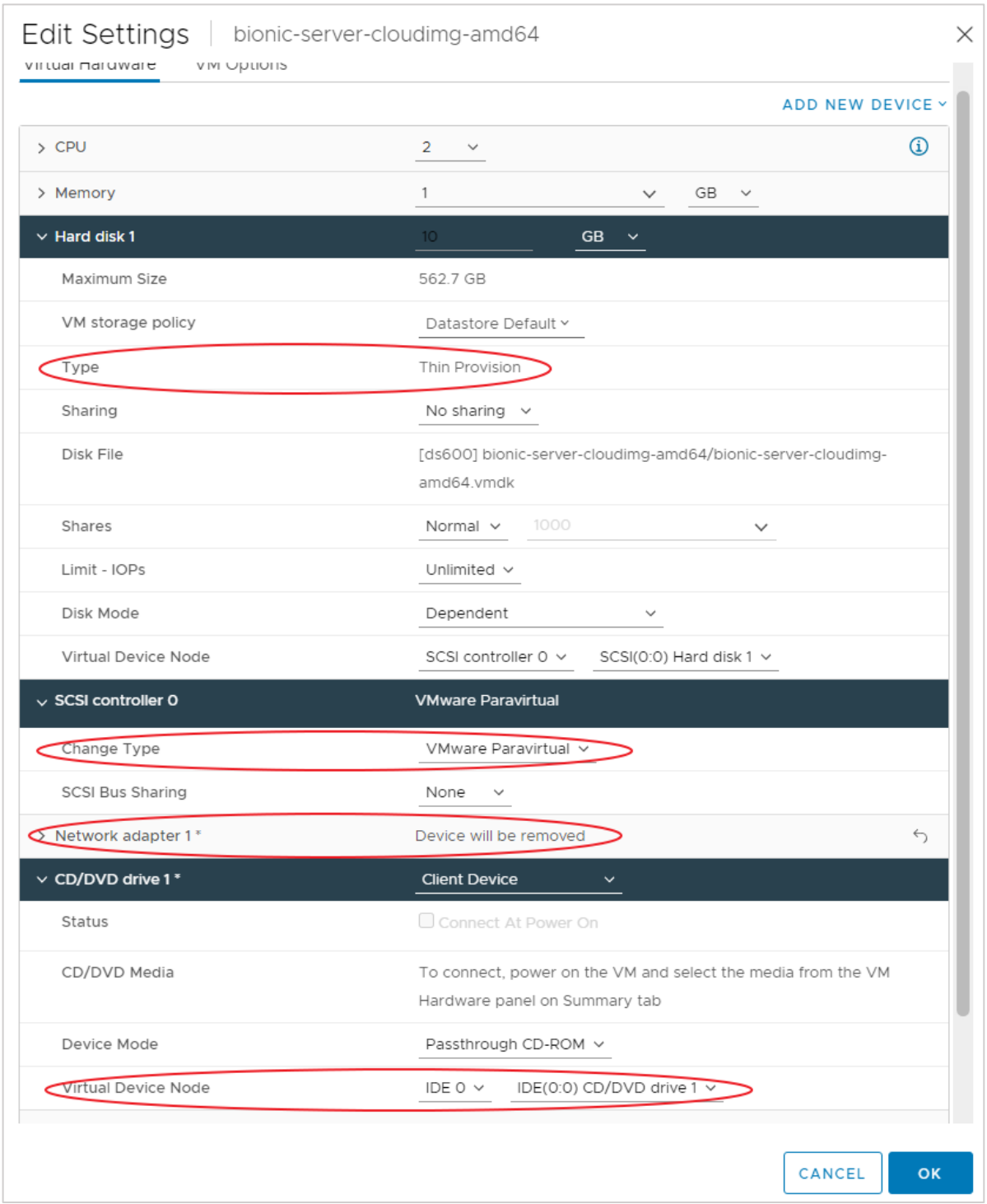

Рисунок 78

Запускаем получившуюся ВМ. Вводим установленный пароль (пользователь ubuntu), входим в систему. Потребуется сменить пароль. Меняем на любой.

Затем:

- изменяем файл cloud.cfg;

Cloud-init config может находиться в двух местах:

/etc/cloud/cloud.cfg

/etc/cloud/cloud.cfg.d/*.cfg

- закомментируем секцию users:
- внизу допишем секцию datasource (***Рисунок 79***);

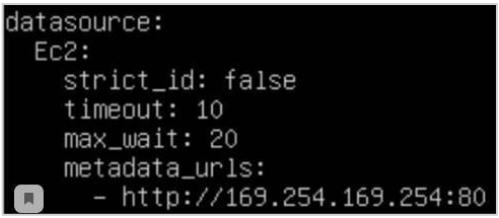

Рисунок 79

- запустим sudo dpkg-reconfigure cloud-init. Запуск команды открывает интерфейс, в котором можно включить/отключить секции datasource;
- отключим всё, оставим EC2 (***Рисунок 80***):

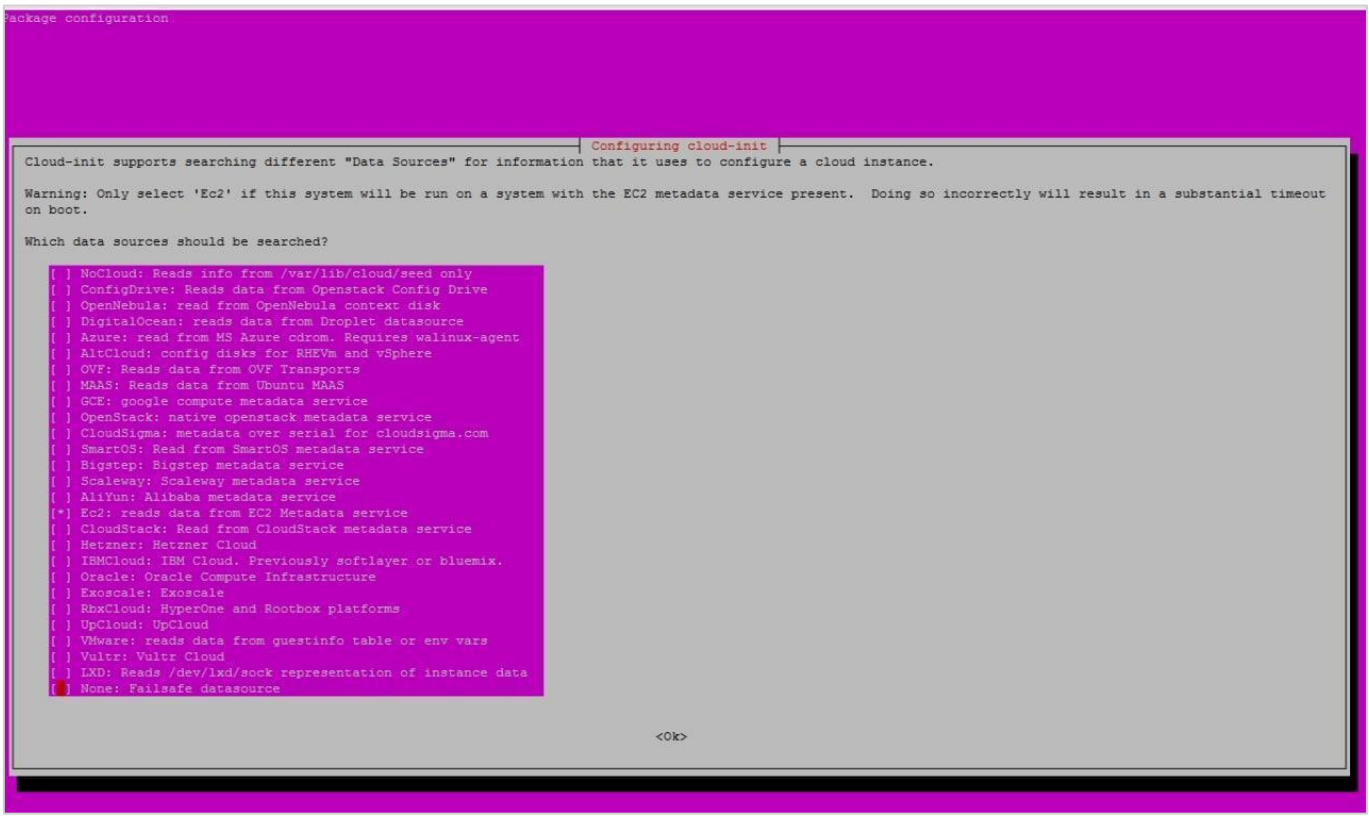

Рисунок 80

- делаем sudo cloud-init clean;
- делаем sudo userdel -f ubuntu;
- отключаем ВМ.

Конвертируем ВМ в шаблон (***Рисунок 81***).

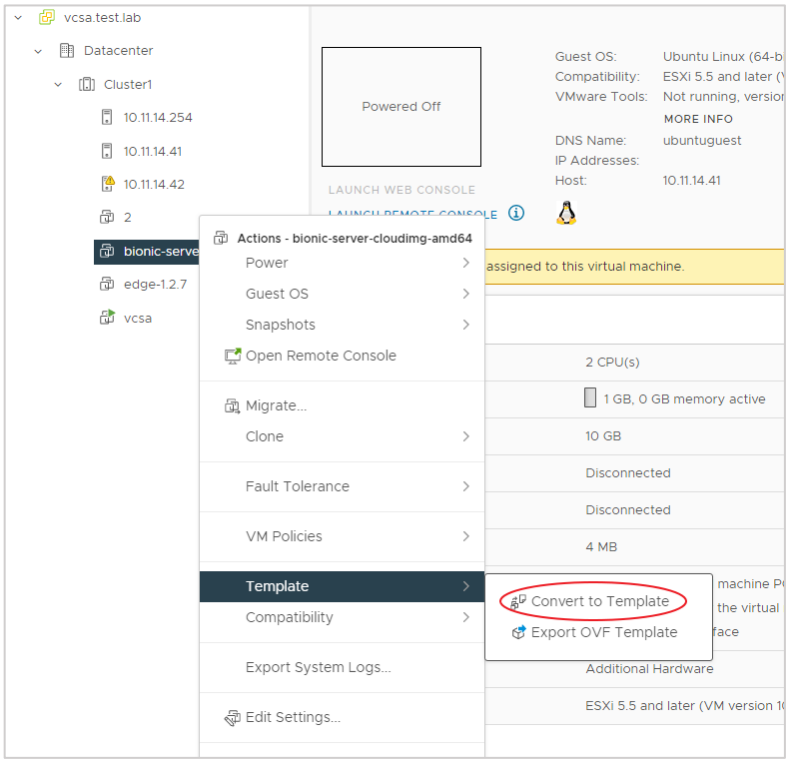

Рисунок 81

После этого необходимо завести шаблон в РУСТЭК-ЕСУ (***Рисунок 82*** – ***Рисунок 84***). Процедура аналогична заведению шаблона для KVM-сегмента (см. ***раздел 6***), необходимо только выбрать другой сегмент (VMware) и другой ID шаблона (выбрать созданный на предыдущих шагах шаблон из списка).

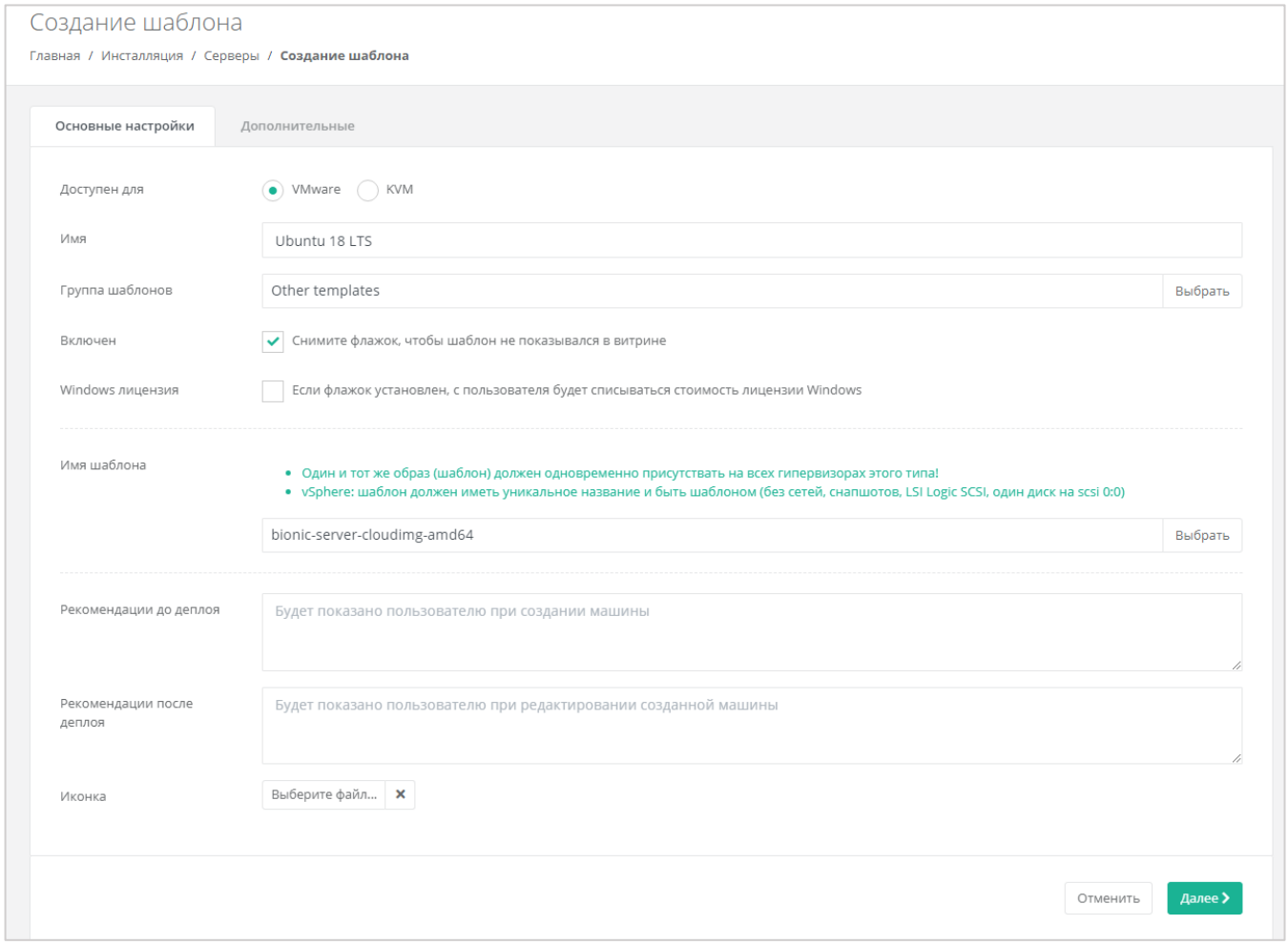

Рисунок 82

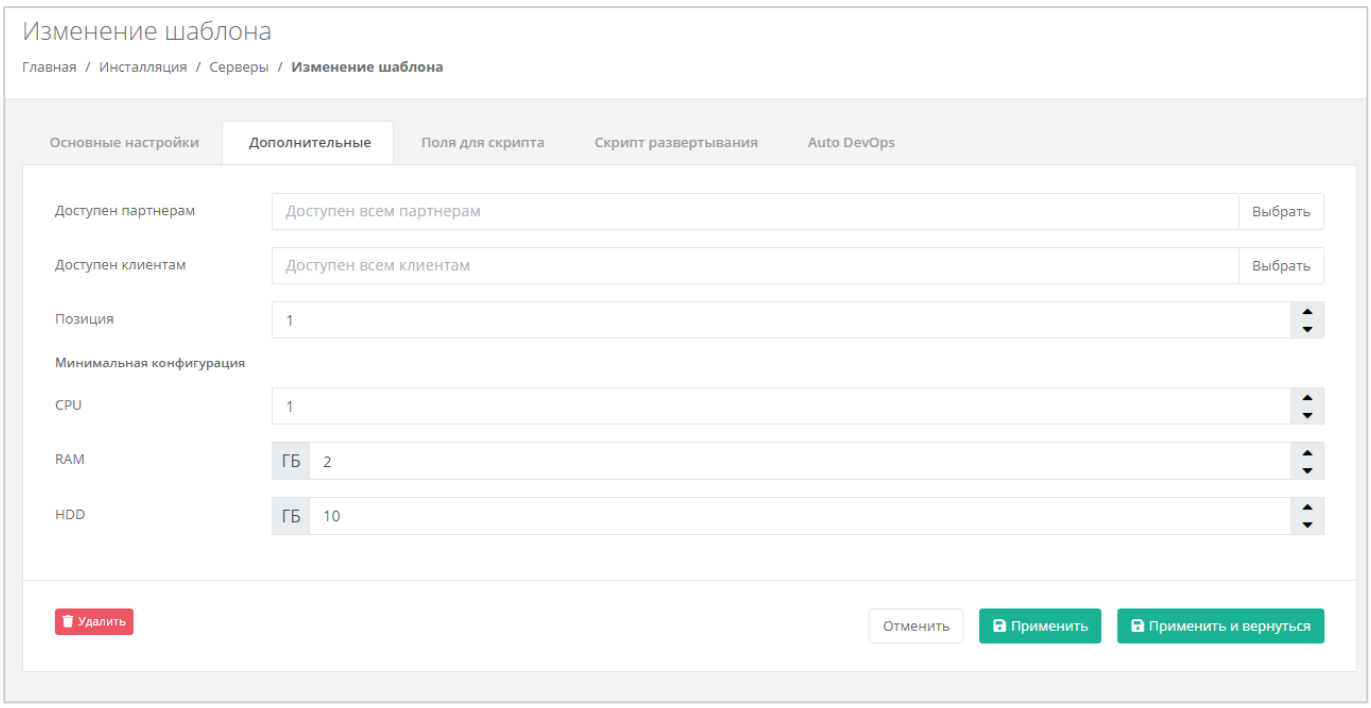

Рисунок 83

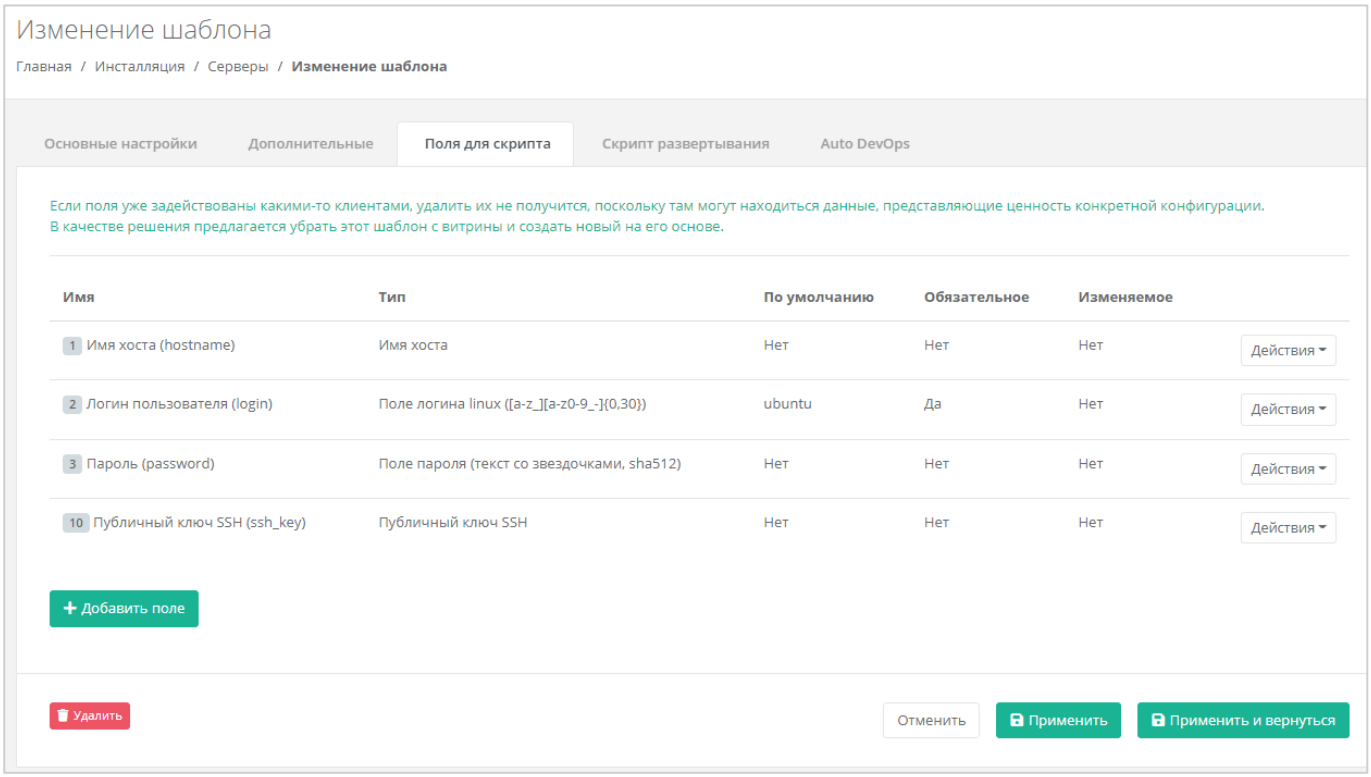

Рисунок 84

Далее во вкладке ***Скрипт развёртывания*** необходимо добавить скрипт развёртывания.

Скрипт развёртывания применяется во время развёртывания виртуальной машины внутри операционной системы сервера.

Примечание: универсальный скрипт развёртывания для Linux OS приложен ниже в документации в разделе Универсальный скрипт развёртывания.

На вкладке ***Auto DevOps*** можно настроить Auto DevOps-скрипт. Скрипт обращается к API РУСТЭК-ЕСУ для выполнения указанных в скрипте операций.

Auto DevOps-скрипт пишется на языке Python и используется для выполнения дополнительных операций с сервером во время его создания и/или запуска.

Примечание: внесение изменений в Auto DevOps-скрипт рекомендуется только для вендоров. Просьба не редактировать настройки скрипта самостоятельно.

Пример скрипта приведён в Приложении 1.

!!!Важно!!! После внесения изменений в скрипт нужно обязательно нажать кнопку Применить.

В результате редактирования настроек Auto DevOps-скрипта вносятся изменения в панели управления. Например, применяются необходимые шаблоны брандмауэра после разворачивания виртуальной машины.

После внесения изменений нажимаем кнопку **Применить и вернуться**. Созданный шаблон ВМ появится в списке шаблонов и из него можно будет создавать ВМ.

9. Добавление ресурсных пулов партнёру

После того как ресурсные пулы для обоих сегментов были настроены их необходимо добавить партнёру.

Переходим в **Администрирование** – **Партнёры**. Выбираем созданного партнёра, переходим на вкладку ***Основные настройки***. В поле «Ресурсные пулы» выбираем необходимый ресурсный пул (***Рисунок 85***)***.***

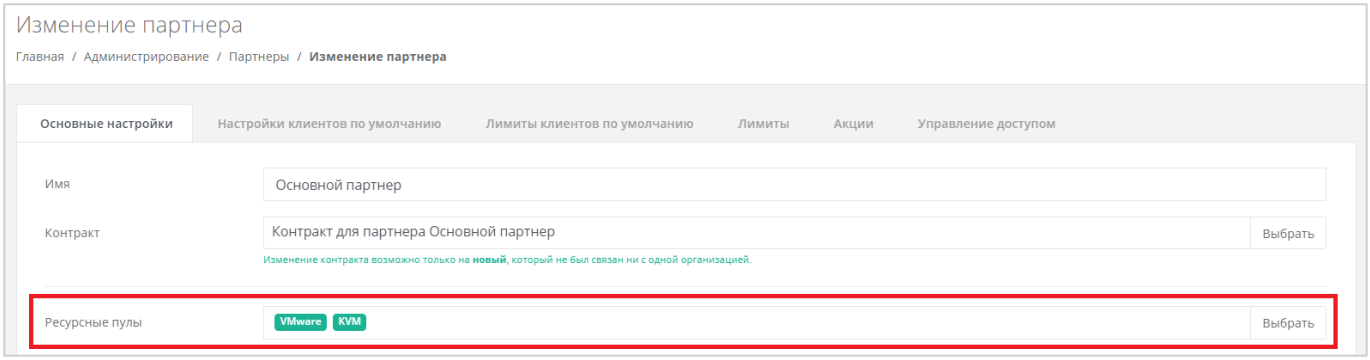

Рисунок 85

Для сохранения настроек нажимаем «Изменить».

10. Создание ВЦОДов в сегментах

Теперь можно создать первые ВЦОДы в сегментах для проверки работоспособности Системы. Сначала в меню **Администрирование** – **Клиенты**, потребуется создать клиента (***Рисунок 86***).

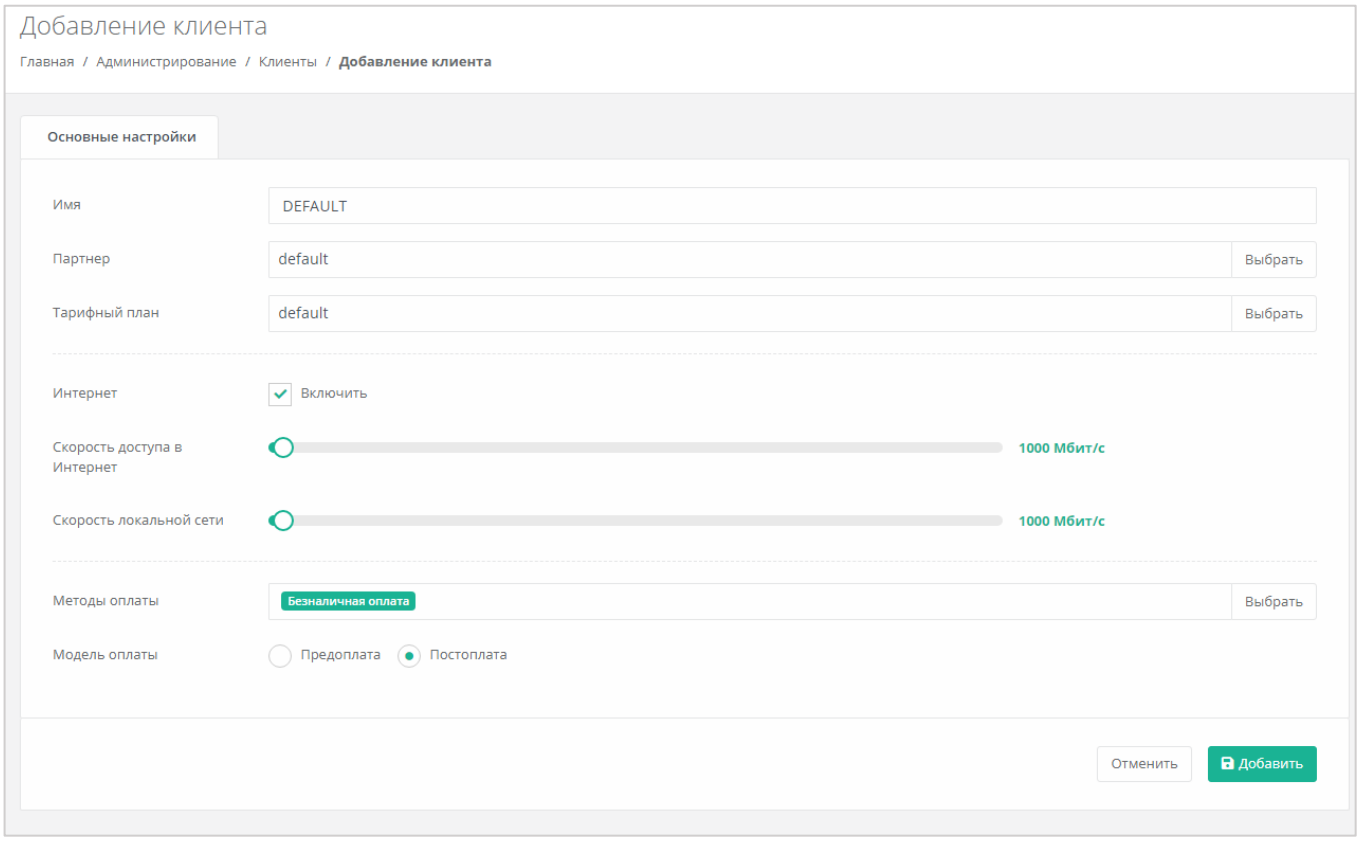

Рисунок 86

В горизонтальном меню нажимаем кнопку «Создать проект» (***Рисунок 87***) и создаём первый проект (***Рисунок 88***).

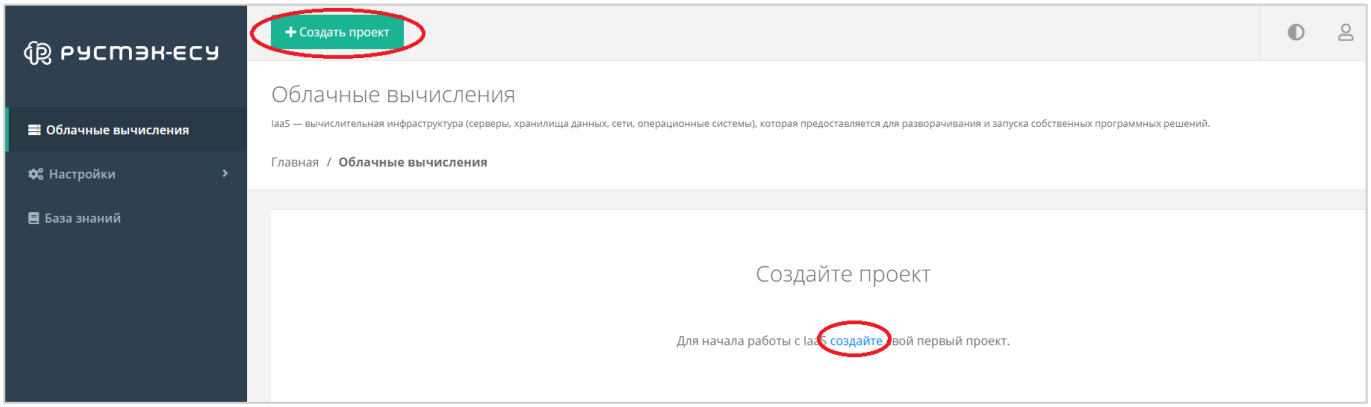

Рисунок 87

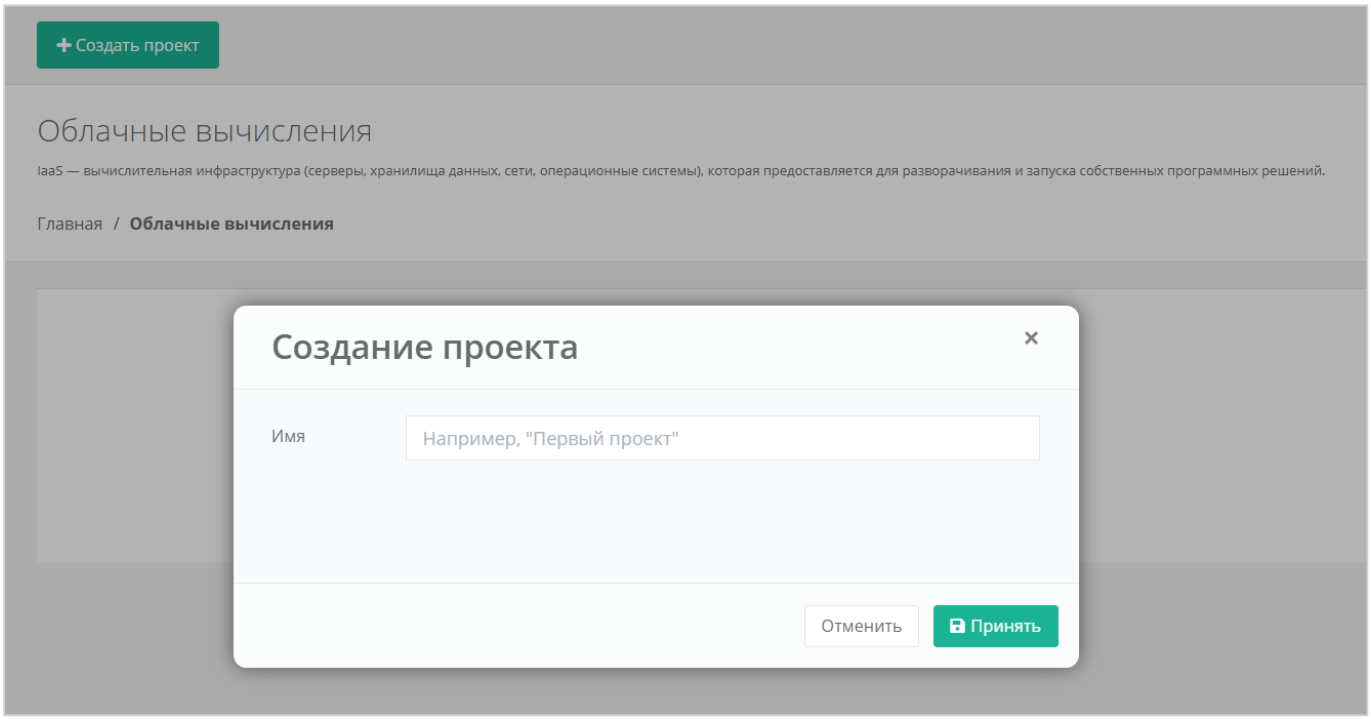

Рисунок 88

Далее переходим в раздел меню **Облачные вычисления** и активируем один из ВЦОДов, например VMware (***Рисунок 89***).

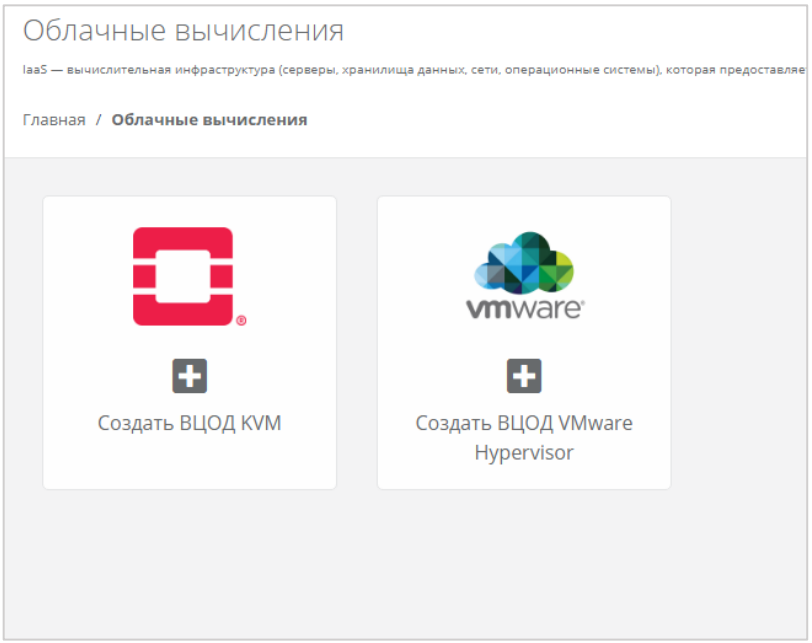

Рисунок 89

После некоторого времени ВЦОД будет готов и иметь статус «работает». В нем можно будет создать виртуальную машину (***Рисунок 90*).**

В нашем примере создано по одному ВЦОД в каждом сегменте (VMware и KVM).

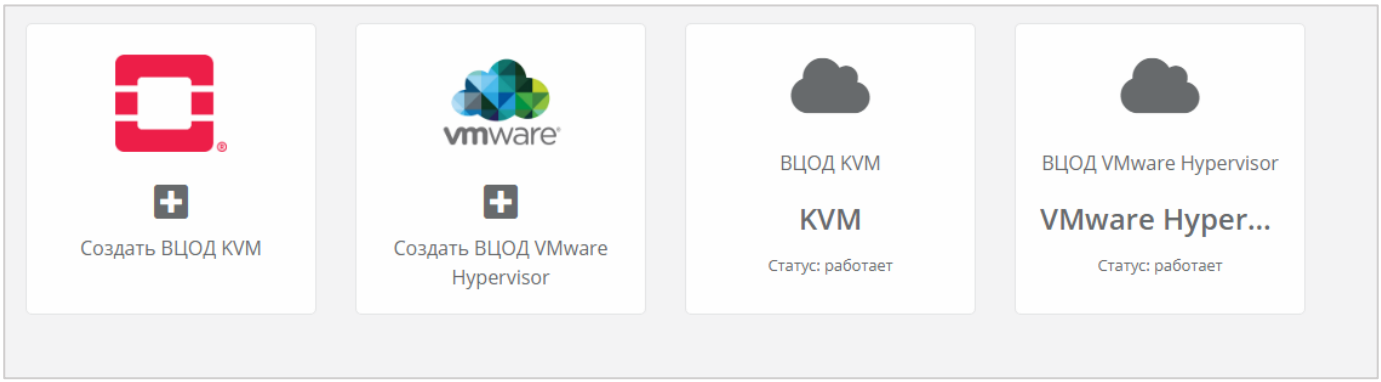
Создать ВЦОД KVM
Создать ВЦОД VMware Hypervisor
ВЦОД KVM
KVM
Статус: работает
ВЦОД VMware Hypervisor
VMware Hyper...
Статус: работает

Рисунок 90

11. Настройка РУСТЭК-ЕСУ для работы с кластерами Kubernetes

11.1. Создание шаблонов Kubernetes для сегмента VMware vSphere

Для разворачивания кластеров Kubernetes в РУСТЭК-ЕСУ сначала необходимо подготовить шаблоны master-ноды, с которой будет происходить управление кластером и обычной ноды.

Сначала необходимо скачать подготовленные нашей командой шаблоны в архивах.

Master-нода: https://ncl.sbcloud.ru/s/EtYnDbQqoe2xPMF/download/k8s-1.22.1.master.zip

Нода: https://ncl.sbcloud.ru/s/HHQwGfJQso3M7ZR/download/k8s-1.22.1.node.zip

Распаковываем архивы.

Заходим в панель управления VMware vSphere. В панель управления необходимо загрузить распакованные образы. Для этого выбираем директорию, в которую будут загружены образы, в нашем случае это ESU3-Test, кликаем по ней правой кнопкой мыши и выбираем «Deploy OVF Template» (***Рисунок 91***).

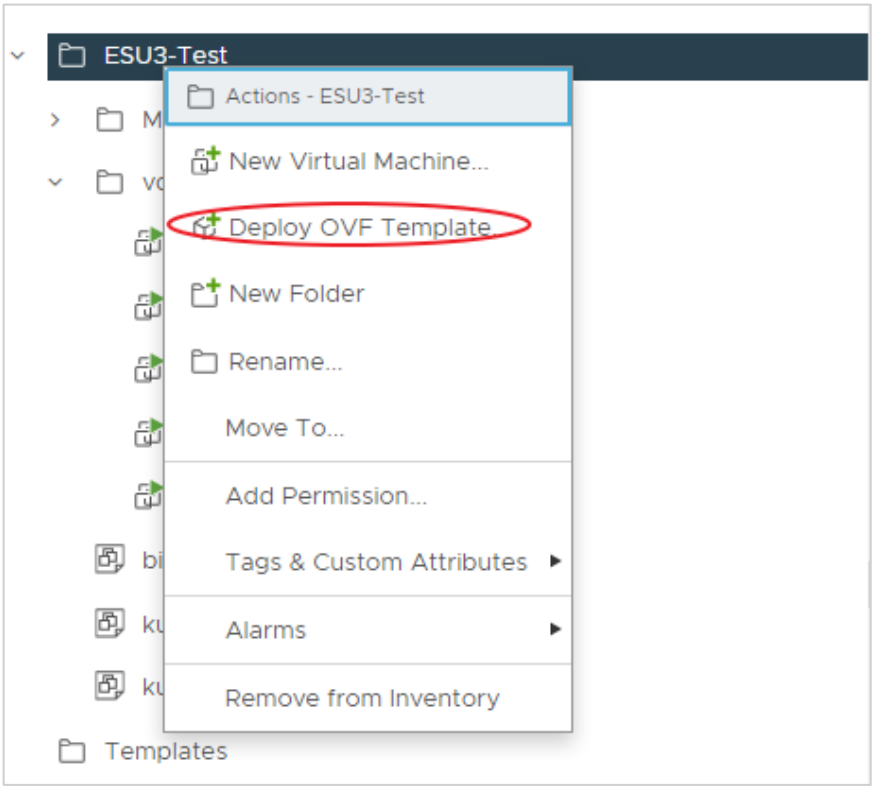

Рисунок 91

Далее выбираем загрузку файла с локального компьютера (***Рисунок 92***).

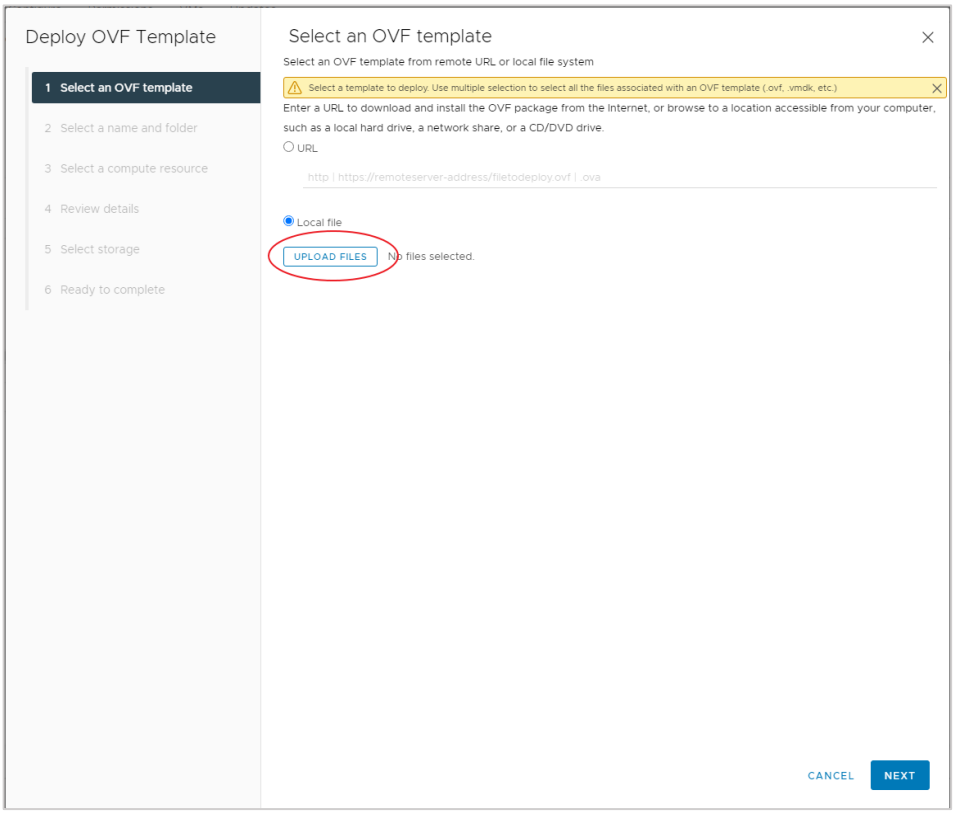

Рисунок 92

В открывшемся окне выбираем файлы нашего образа. После выбора файлов нажмём кнопку «Next» (***Рисунок 93***).

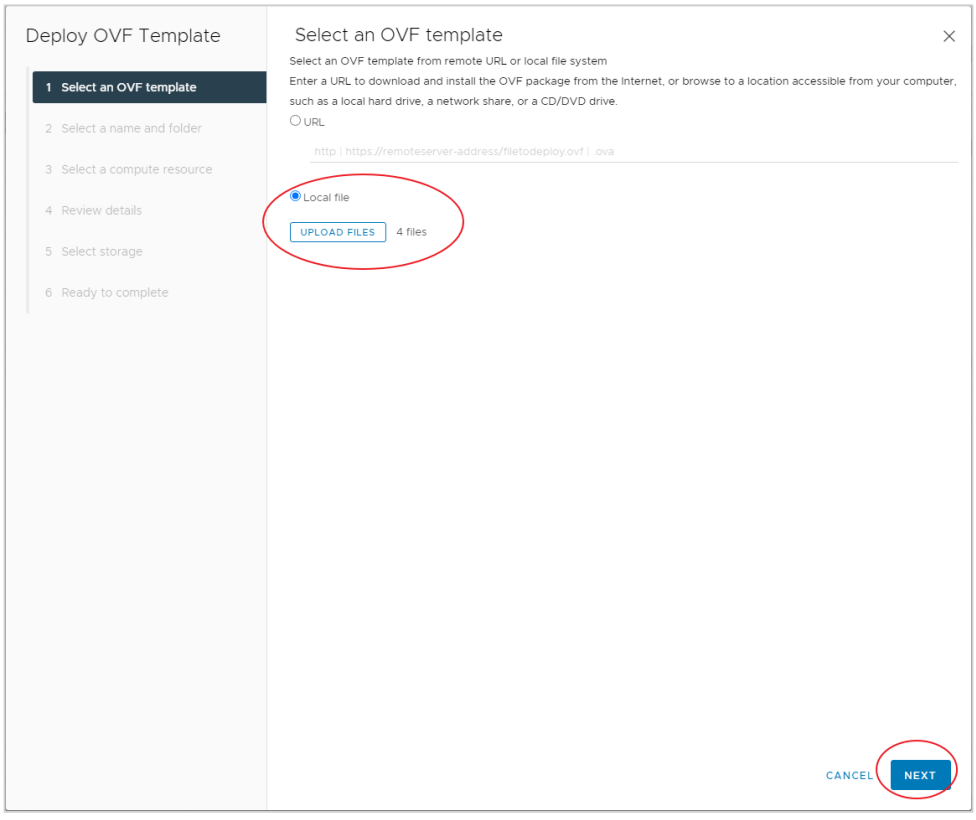

Рисунок 93

Выберем название шаблона и папку для хранения (***Рисунок 94***).

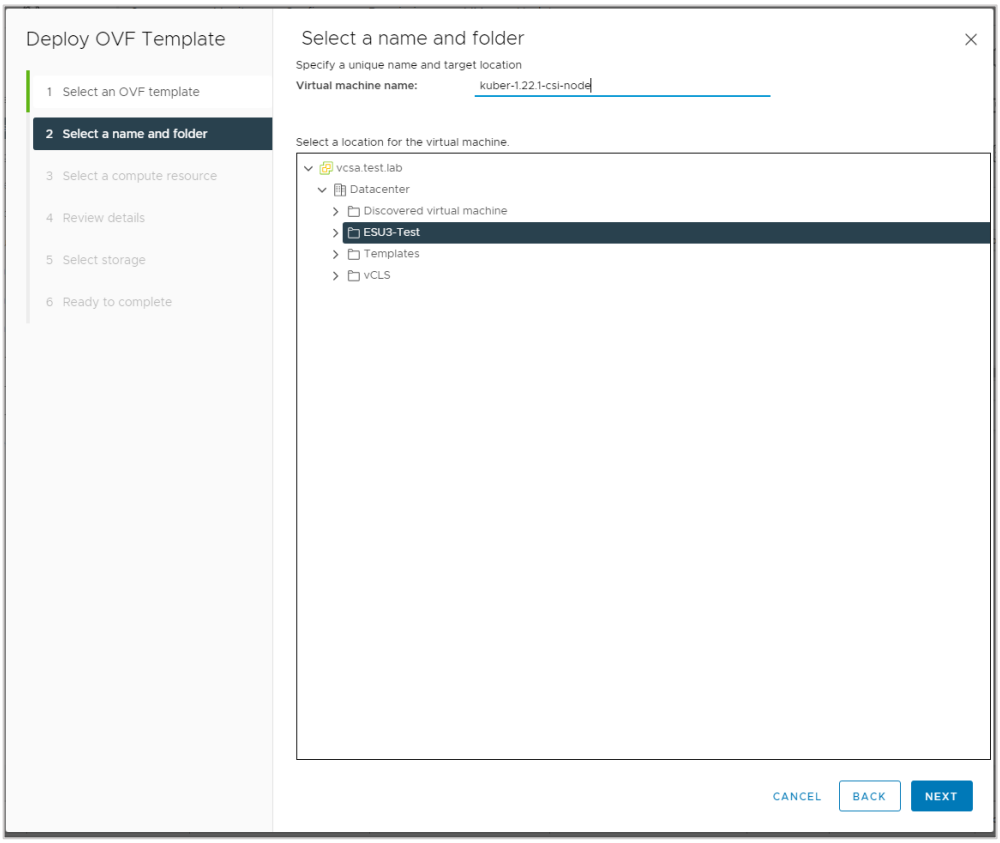

Рисунок 94

Выберем кластер, где будет храниться шаблон и нажимаем «NEXT» (***Рисунок 95***).

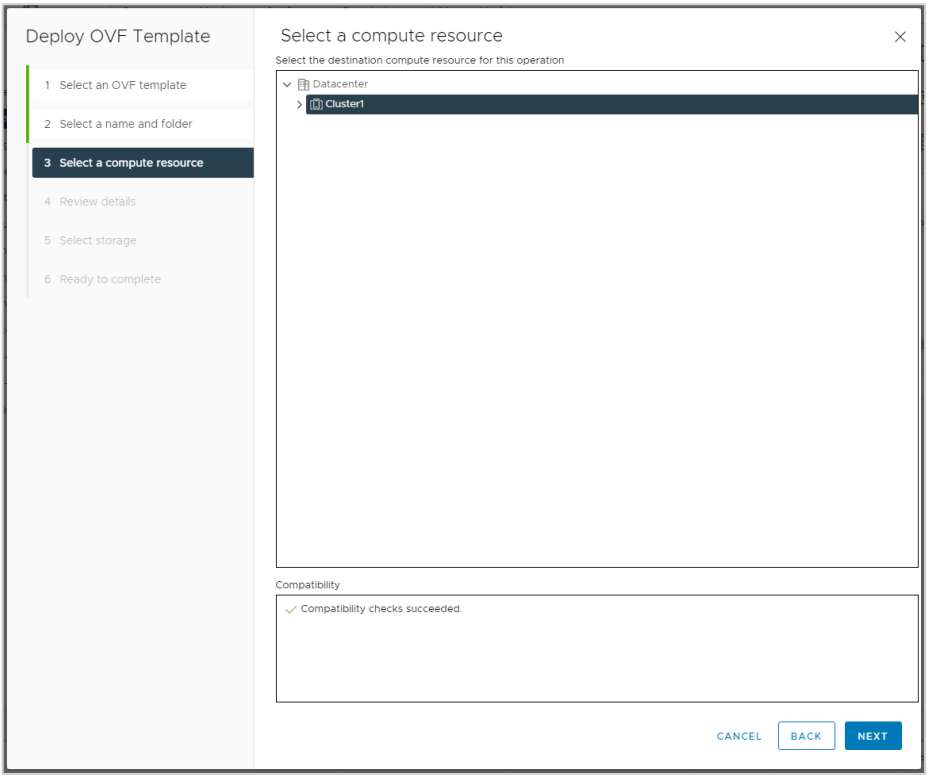

Рисунок 95

Нажимаем «NEXT» (***Рисунок 96***).

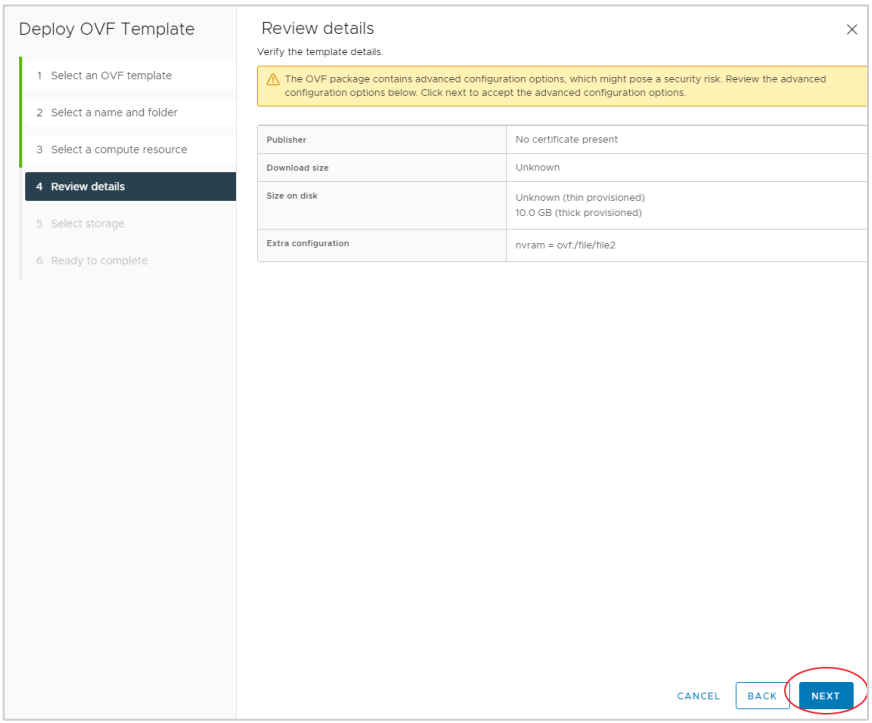

Рисунок 96

Выбираем датастор для хранения шаблона (***Рисунок 97***).

Обязательно выбираем формат диска Thin Provision!

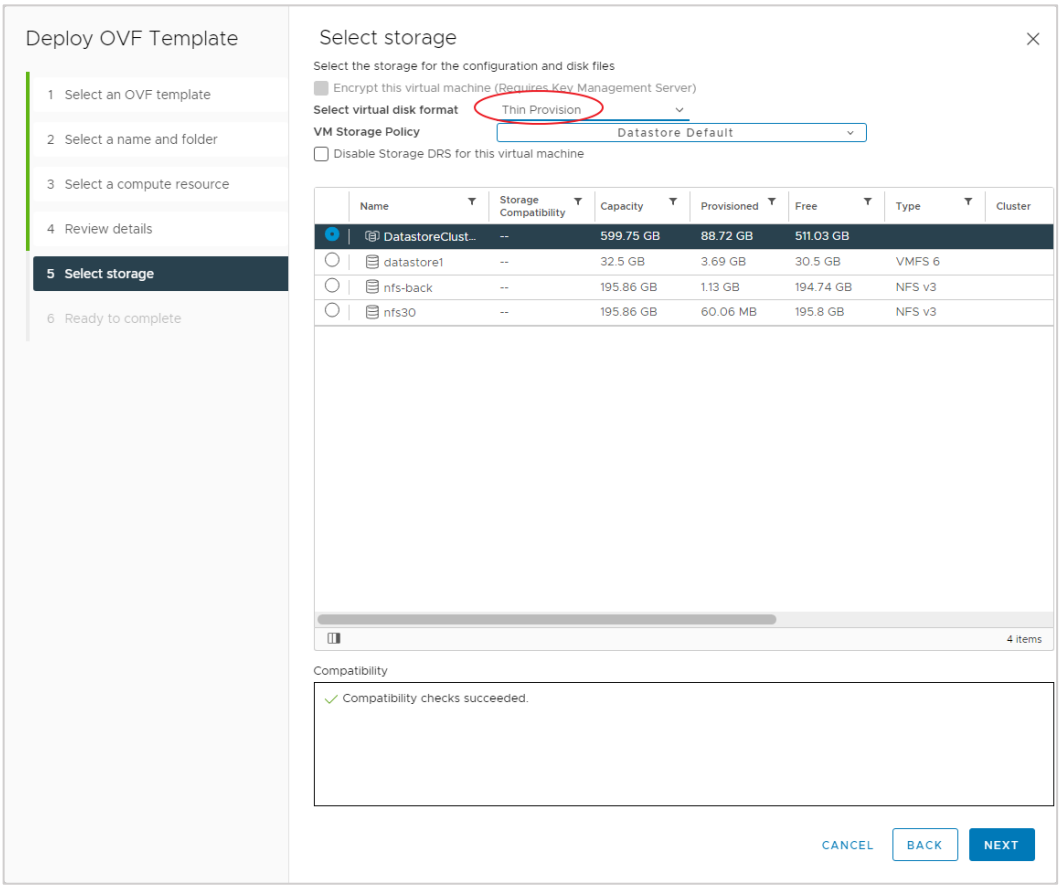

Рисунок 97

Завершаем процесс нажатием кнопки «FINISH» в открывшемся окне (***Рисунок 98***).

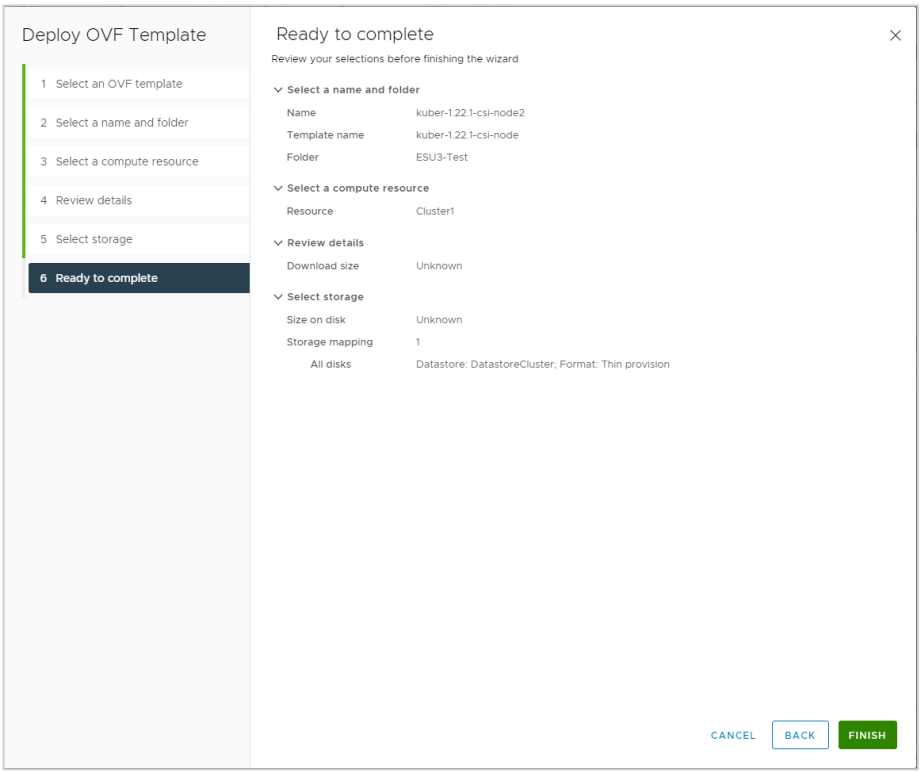

Рисунок 98

После успешной загрузки необходимо сконвертировать созданную таким образом ВМ в темплейт. Для этого нажмём по ней правой кнопкой мыши и выберем «Template» –«Convert to Template» (***Рисунок 99***).

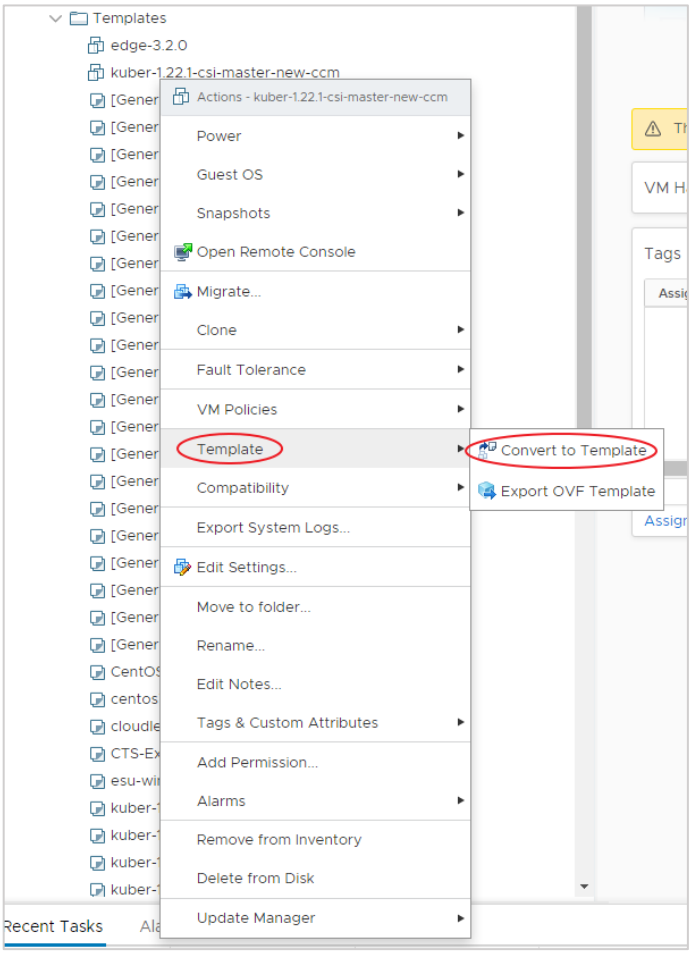

Рисунок 99

Данную операцию по загрузке и конвертации необходимо проделать для шаблона master-ноды и для обычной ноды!

После успешной загрузки шаблонов в VMware vSphere необходимо настроить РУСТЭК-ЕСУ для работы с ними. Для этого в панели управления РУСТЭК-ЕСУ переходим в меню **Инсталляция** – **Шаблоны** – **Kubernetes** и нажимаем «Создать шаблон» (***Рисунок 100***).

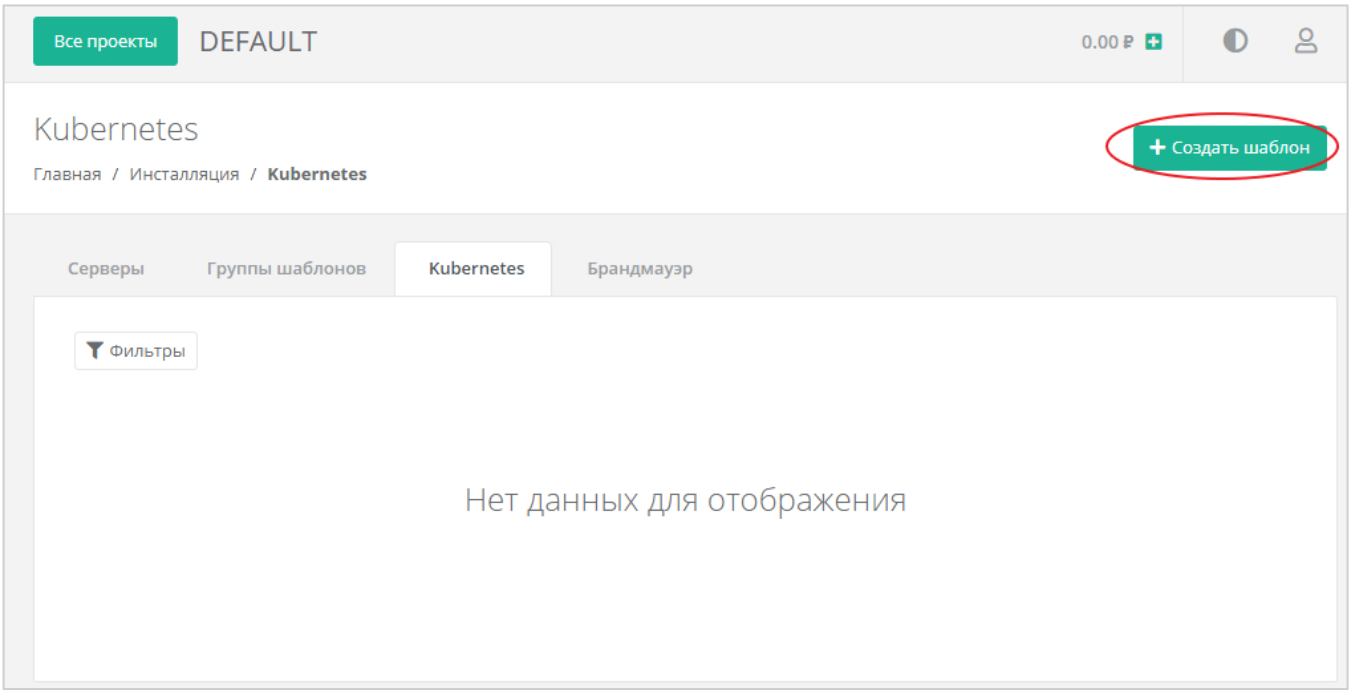

Рисунок 100

В открывшемся окне заполняем следующие параметры (***Рисунок 101***):

- Доступен для – выбираем сегмент VMware.
- Имя – произвольное имя.
- Включен – чек-бокс установлен.
- Темплейт мастера – выбираем шаблон мастера, загруженный в vSphere из выпадающего списка.
- Темплейт ноды – выбираем шаблон ноды, загруженный в vSphere из выпадающего списка.
- Видимый шаблон ОС – выбираем любой шаблон из списка (влияет только на название, которое будет отображаться в списке серверов).
- Минимальная конфигурация – рекомендуемая конфигурация для наших шаблонов: 2vCPU, 2RAM, 10ГБ HDD.

Нажимаем «Применить».

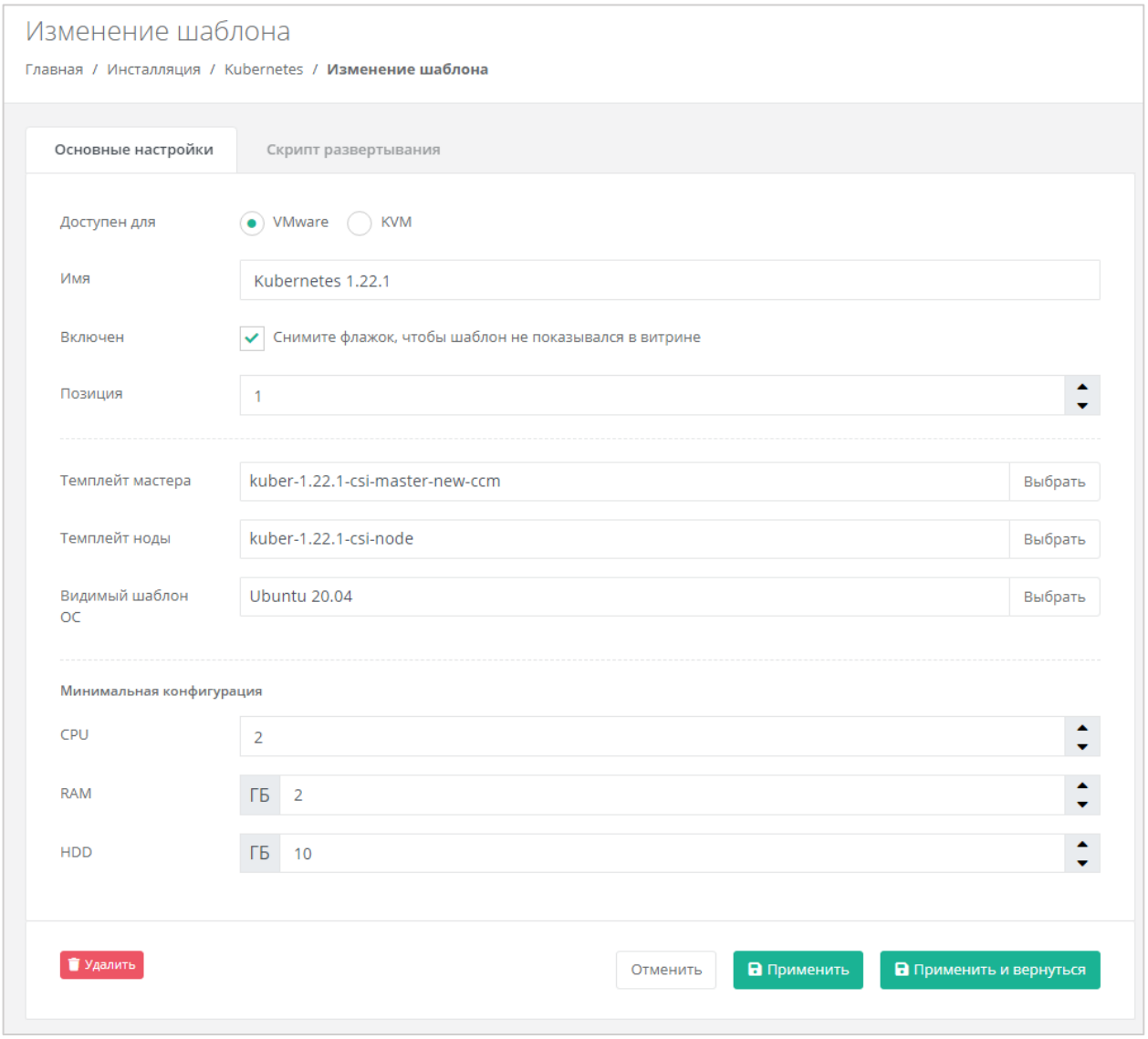

Рисунок 101

Далее во вкладке ***Скрипт развёртывания*** необходимо добавить скрипт.

Скрипт развёртывания:

```
from authentication.models import PubKey, Token


def get_metadata(master=None, node=None):
    if master:
        return _prepare_master(master)
    else:
        return _prepare_node(node)


def _prepare_master(master):
    hypervisor = master.vdc.hypervisor
    api_url = hypervisor.get_setting('platform_internal_url')
    api_token = hypervisor.get_setting('edge_api_token')

    sa_token = Token(user=master.service_user)
    sa_token.save()
    sa_token = sa_token.original_key

    return {
        'user_data': f"""\
#cloud-config
debug:
  verbose: true
cloud_init_modules:
  - migrator
  - seed_random
  - bootcmd
  - write-files
  - growpart
  - resizefs
  - set_hostname
  - update_hostname
  - update_etc_hosts
  - users-groups
  - ssh
  - runcmd
runcmd:
  - runner install --api_url="{api_url}" --token="{api_token}" --sa_token="{sa_token}" --
runner_id="{master.short_id}" --ifname=eth0 --kubernetes_uuid="{master.id}" --version="1.22.1"
fqdn: "{master.master_hostname}"
```

```
manage_etc_hosts: true
disable_root: false
ssh_pwauth: yes
users:
  - default
ssh_authorized_keys:
  - ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDKZnwlDIoHsfZukwf/QnHP8KR/diFMQgLFxG0Doe9qdZ/nE7
xf3bUF9WNXwMEemQv6Vo6Jdp0kTswT+ZuELlxcvd4OgnIBChdY8qym/4/BFMqFJz6IJ1Bhenp/+bvy/cW
R2bBKNiYb0Cw5dWU+0xbS75l6jy0oH3zCwVTNGQ7ieB5cwJaq3w9LYuXGITUN6pko3mJKMhQ1JB7mr
e8ZGkzKIwux5Eut4me1JCFfi/bGFlUUB/uFkzJIHtv4nlAmz3pW+Wv/6eqXXoaBrGp9Dmp3qPmnXtAywsn
KGZ6ohp2jIcmJZ69ceJvB1jx5IoIR9W+ntBwlVhvmOdkSVy4yHiGL deploy@localhost
chpasswd:
  expire: false
  list:
    - root:
timezone: "Europe/Moscow"
package_update: false
datasource:
  Ec2:
    strict_id: false
    timeout: 5
    max_wait: 5
    metadata_urls:
      - http://169.254.169.254:80
""",
        'hostname': master.master_hostname[:15],
        'instance-id': master.short_id,
    }

def _prepare_node(node):
    pub_keys = [node.kubernetes.service_public_key, node.kubernetes.user_public_key]
    pub_keys = '\n'.join([f'  - "{k}"' for k in pub_keys])

    internal_ip = node.ports[0].ip_address

    return {
        'user_data': f"""\
#cloud-config
debug:
  verbose: true
```

```
cloud_init_modules:
  - seed_random
  - bootcmd
  - write-files
  - growpart
  - resizefs
  - set_hostname
  - users-groups
  - ssh
bootcmd:
  - echo {internal_ip} {node.hostname or node.short_id[:15]} > /etc/hosts
  - echo "127.0.0.1 localhost" >> /etc/hosts
disable_root: false
fqdn: "{node.hostname or node.short_id[:15]}"
ssh_pwauth: yes
users:
  - default
ssh_authorized_keys:
{pub_keys}
chpasswd:
  expire: false
  list:
    - root:
timezone: "Europe/Moscow"
package_update: false
datasource:
  Ec2:
    strict_id: false
    timeout: 5
    max_wait: 5
    metadata_urls:
      - http://169.254.169.254:80
""",
        'hostname': node.short_id[:15],
        'instance-id': node.short_id,
    }
```

После установки скрипта развёртывания нажимаем «Применить и вернуться» (***Рисунок 102***).

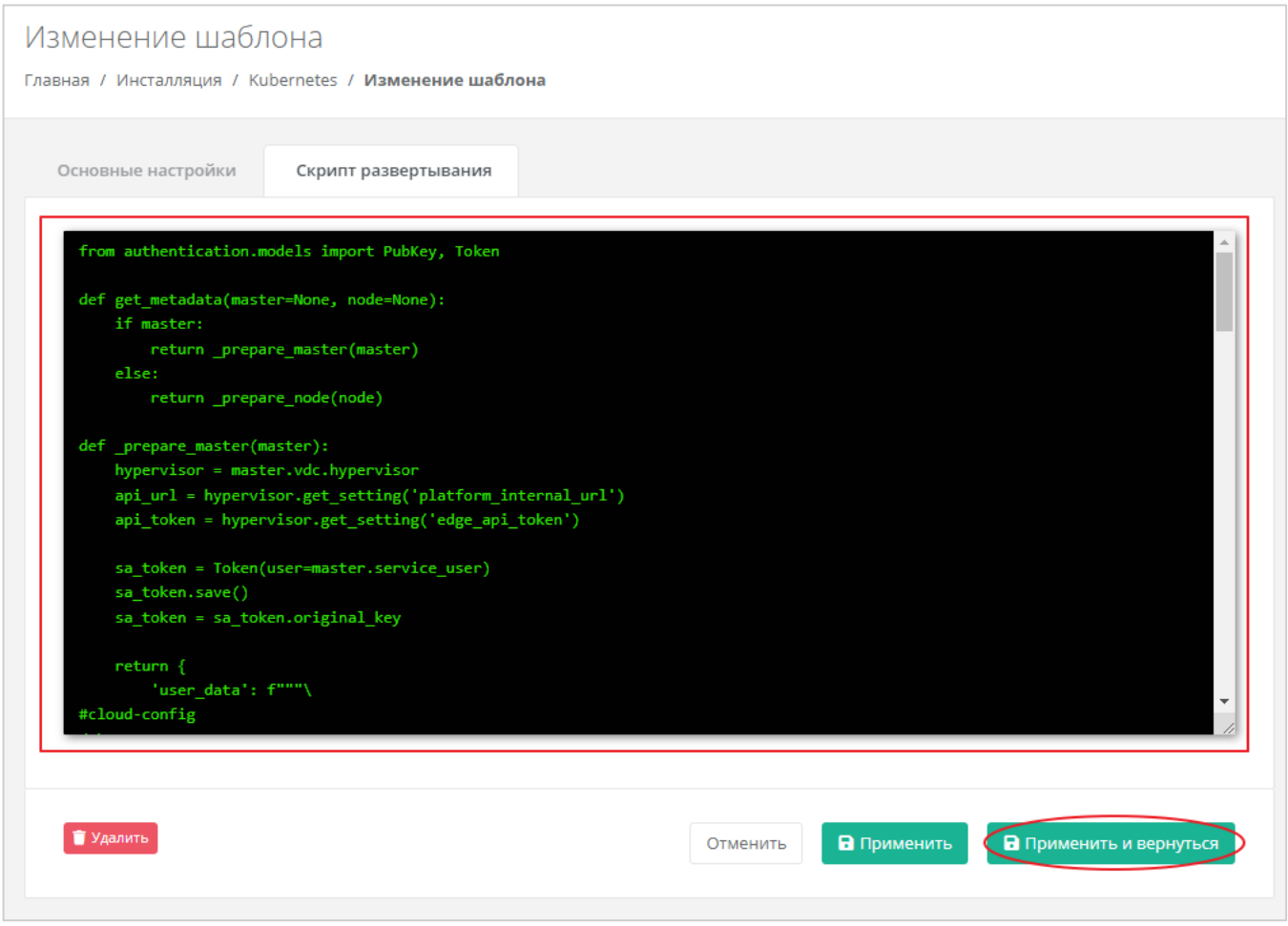

Рисунок 102

На этом настройка шаблона завершена, и он отобразится в списке шаблонов Kubernetes (***Рисунок 103***), а также будет доступен для создания в меню **Кластеры Kubernetes** для пользователя (***Рисунок 104***).

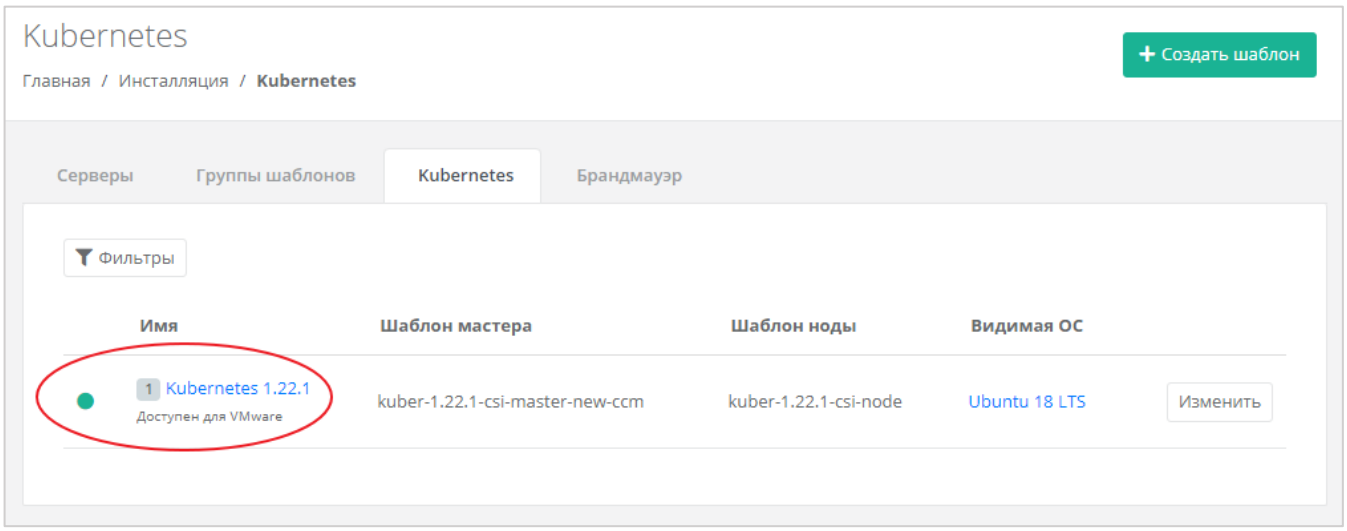

Рисунок 103

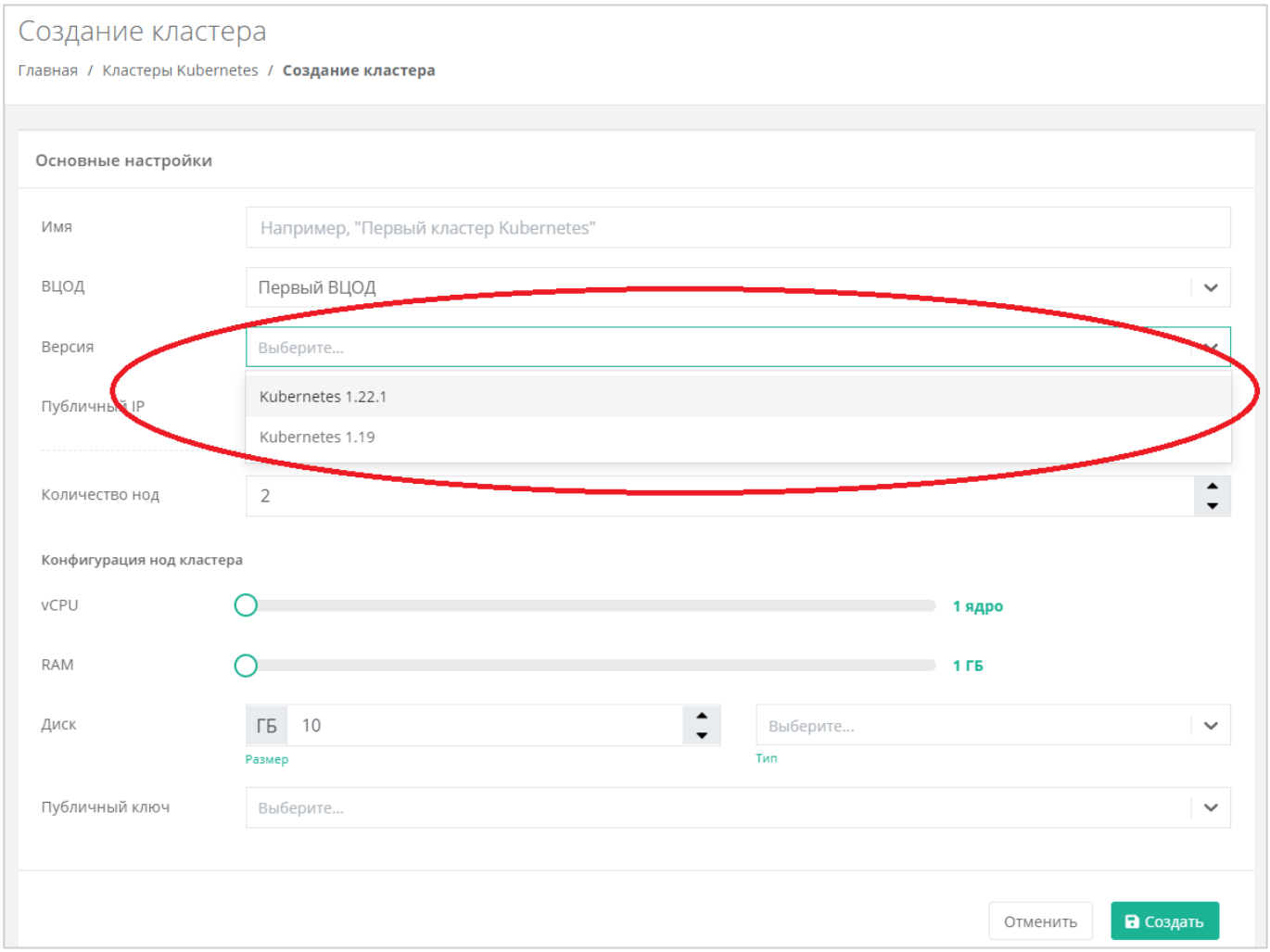

Рисунок 104

11.2. Создание шаблонов Kubernetes для сегмента РУСТЭК/KVM

Для создания шаблонов Kubernetes в сегменте РУСТЭК необходимо зайти по SSH (root:rustack) на один из контроллеров РУСТЭК, скачать .vmdk образы master-ноды и ноды, конвертировать их в формат .raw и создать из них images.

Скачиваем образы в директорию **/tmp**, используя указанные в команде ссылки:

```
cd /tmp


curl -o kuber-1.22.1-csi-node-1.vmdk
https://ncl.sbcloud.ru/s/L3j8SNFKrkqcHjJ/download/kuber-1.22.1-csi-node-1.vmdk


curl -o kuber-1.22.1-csi-master-new-ccm-1.vmdk
https://ncl.sbcloud.ru/s/9HqHasftppNM4iq/download/kuber-1.22.1-csi-master-new-ccm-
1.vmdk
```

Конвертируем образы в формат .raw:

```
qemu-img convert -p -O raw /tmp/kuber-1.22.1-csi-master-new-ccm-1.vmdk /tmp/kuber-1.22.1-csi-master-new-ccm-1.raw


qemu-img convert -p -O raw /tmp/kuber-1.22.1-csi-node-1.vmdk /tmp/kuber-1.22.1-csi-node-1.raw
```

Удаляем исходники (.vmdk):

```
rm /tmp/kuber-1.22.1-csi-node-1.vmdk

rm /tmp/kuber-1.22.1-csi-master-new-ccm-1.vmdk
```

Создаём images (***Рисунок 105***):

```
openstack image create --disk-format raw --container-format bare --public --property hw_disk_bus=scsi --property hw_scsi_model=virtio-scsi --property hw_vif_model=virtio --property image_type=master --file /tmp/kuber-1.22.1-csi-master-new-ccm-1.raw kuber-1.22.1-csi-master-new-ccm


openstack image create --disk-format raw --container-format bare --public --property hw_disk_bus=scsi --property hw_scsi_model=virtio-scsi --property hw_vif_model=virtio --property image_type=master --file /tmp/kuber-1.22.1-csi-node-1.raw kuber-1.22.1-csi-node
```

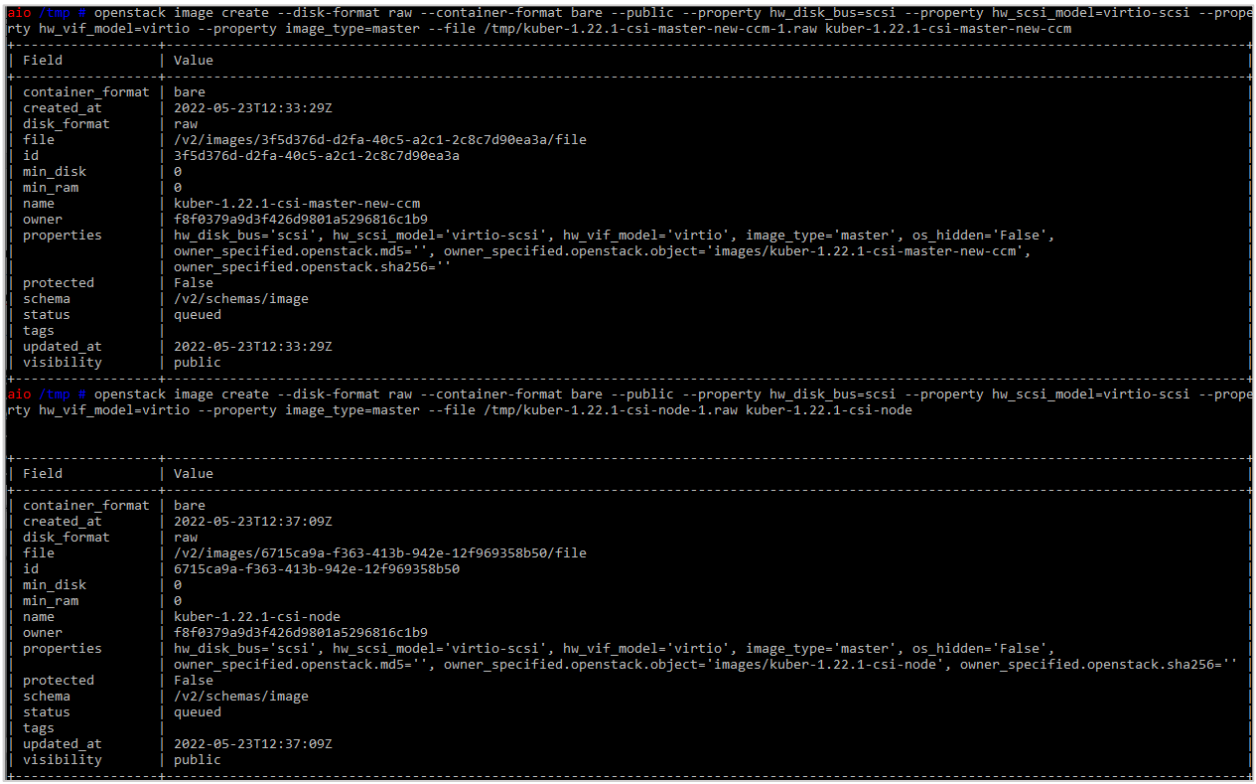

Рисунок 105

Удаляем образы (.raw):

```
rm /tmp/kuber-1.22.1-csi-node-1.raw
```

```
rm /tmp/kuber-1.22.1-csi-master-new-ccm-1.raw
```

После успешной загрузки шаблонов в РУСТЭК необходимо настроить РУСТЭК-ЕСУ для работы с ними. Для этого в панели управления РУСТЭК-ЕСУ переходим в меню **Инсталляция** – **Шаблоны** – **Kubernetes** и нажимаем «Создать шаблон» (***Рисунок 106***).

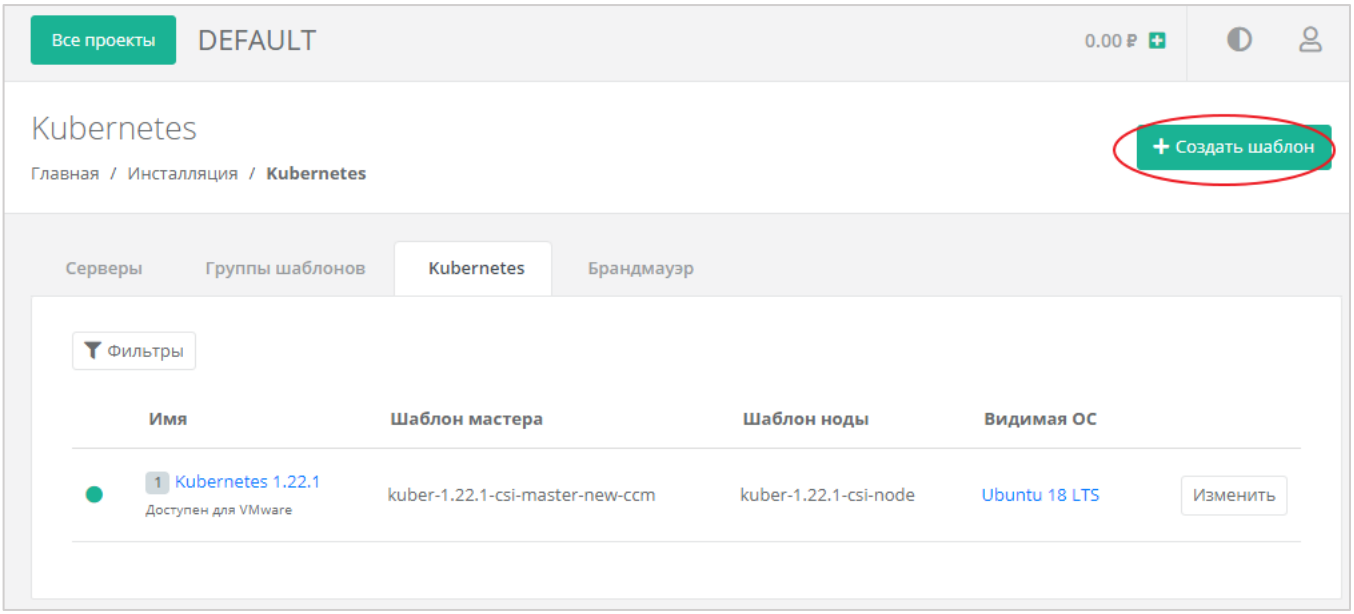

Рисунок 106

В открывшемся окне заполняем следующие параметры (***Рисунок 107***):

- Доступен для – выбираем сегмент KVM.
- Имя – произвольное имя.
- Включен – чек-бокс установлен.
- Темплейт мастера – выбираем шаблон мастера, загруженный в РУСТЭК из выпадающего списка.
- Темплейт ноды – выбираем шаблон ноды, загруженный в РУСТЭК из выпадающего списка.
- Видимый шаблон ОС – выбираем любой шаблон из списка (влияет только на название, которое будет отображаться в списке серверов).
- Минимальная конфигурация – рекомендуемая конфигурация для наших шаблонов: 2vCPU, 2RAM, 10ГБ HDD.

Нажимаем «Применить».

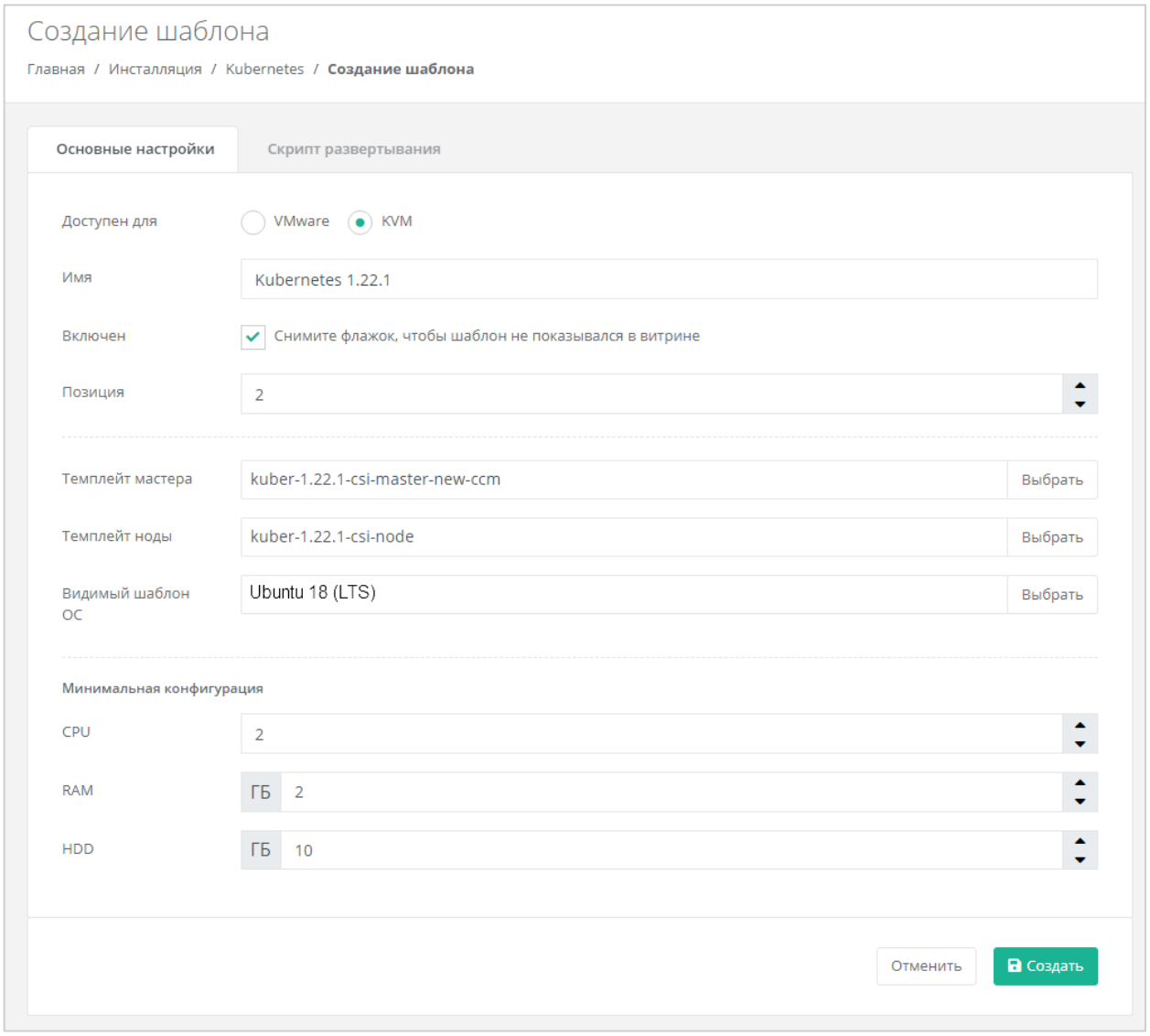

Рисунок 107

Далее во вкладке ***Скрипт развёртывания*** необходимо добавить скрипт.

Скрипт развёртывания:

```
from authentication.models import PubKey, Token


def get_metadata(master=None, node=None):
    if master:
        return _prepare_master(master)
    else:
        return _prepare_node(node)


def _prepare_master(master):
    hypervisor = master.vdc.hypervisor
    api_url = hypervisor.get_setting('platform_internal_url')
    api_token = hypervisor.get_setting('edge_api_token')

    sa_token = Token(user=master.service_user)
    sa_token.save()
    sa_token = sa_token.original_key

    return {
        'user_data': f"""\
#cloud-config
debug:
  verbose: true
cloud_init_modules:
  - migrator
  - seed_random
  - bootcmd
  - write-files
  - growpart
  - resizefs
  - set_hostname
  - update_hostname
  - update_etc_hosts
  - users-groups
  - ssh
  - runcmd
runcmd:
  - runner install --api_url="{api_url}" --token="{api_token}" --sa_token="{sa_token}" --
runner_id="{master.short_id}" --ifname=eth0 --kubernetes_uuid="{master.id}" --version="1.22.1"
fqdn: "{master.master_hostname}"
```

```
manage_etc_hosts: true
disable_root: false
ssh_pwauth: yes
users:
  - default
ssh_authorized_keys:
  - ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDKZnwlDIoHsfZukwf/QnHP8KR/diFMQgLFxG0Doe9qdZ/nE7
xf3bUF9WNXwMEemQv6Vo6Jdp0kTswT+ZuELlxcvd4OgnIBChdY8qym/4/BFMqFJz6IJ1Bhenp/+bvy/cW
R2bBKNiYb0Cw5dWU+0xbS75l6jy0oH3zCwVTNGQ7ieB5cwJaq3w9LYuXGITUN6pko3mJKMhQ1JB7mr
e8ZGkzKIwux5Eut4me1JCFfi/bGFlUUB/uFkzJlHtv4nlAmz3pW+Wv/6eqXXoaBrGp9Dmp3qPmnXtAywsn
KGZ6ohp2jIcmJZ69ceJvB1jx5IoIR9W+ntBwlVhvmOdkSVy4yHiGL deploy@localhost
chpasswd:
  expire: false
  list:
    - root:
timezone: "Europe/Moscow"
package_update: false
datasource:
  Ec2:
    strict_id: false
    timeout: 5
    max_wait: 5
    metadata_urls:
      - http://169.254.169.254:80
""",
        'hostname': master.master_hostname[:15],
        'instance-id': master.short_id,
    }

def _prepare_node(node):
    pub_keys = [node.kubernetes.service_public_key, node.kubernetes.user_public_key]
    pub_keys = '\n'.join([f'  - "{k}"' for k in pub_keys])

    internal_ip = node.ports[0].ip_address

    return {
        'user_data': f"""\
#cloud-config
debug:
  verbose: true
```

```
cloud_init_modules:
  - seed_random
  - bootcmd
  - write-files
  - growpart
  - resizefs
  - set_hostname
  - users-groups
  - ssh
bootcmd:
  - echo {internal_ip} {node.hostname or node.short_id[:15]} > /etc/hosts
  - echo "127.0.0.1 localhost" >> /etc/hosts
disable_root: false
fqdn: "{node.hostname or node.short_id[:15]}"
ssh_pwauth: yes
users:
  - default
ssh_authorized_keys:
{pub_keys}
chpasswd:
  expire: false
  list:
    - root:
timezone: "Europe/Moscow"
package_update: false
datasource:
  Ec2:
    strict_id: false
    timeout: 5
    max_wait: 5
    metadata_urls:
      - http://169.254.169.254:80
""",
        'hostname': node.short_id[:15],
        'instance-id': node.short_id,
    }
```

После установки скрипта развёртывания нажимаем «Применить и вернуться» (***Рисунок 108***).

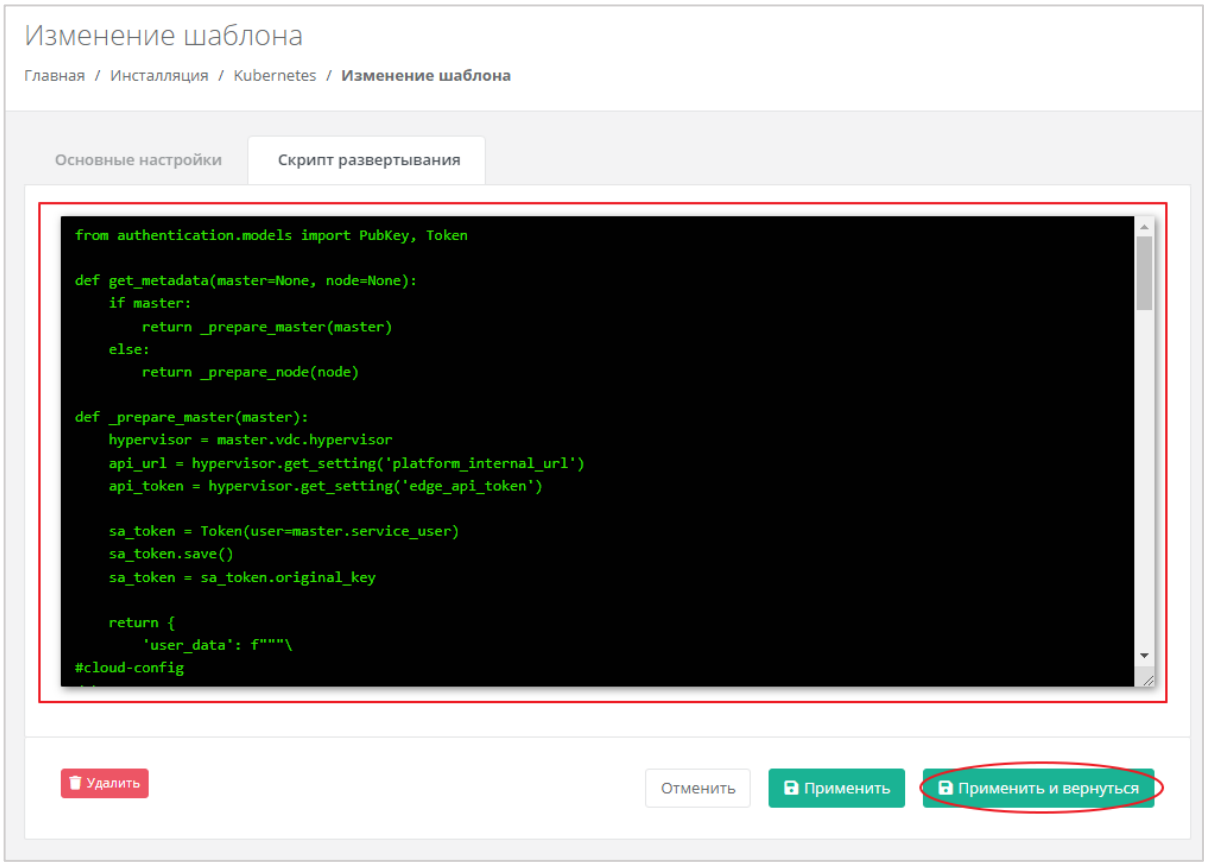

Рисунок 108

На этом настройка шаблона завершена, и он отобразится в списке шаблонов Kubernetes (***Рисунок 109***), а также будет доступен для создания в меню **Кластеры Kubernetes** для пользователя (***Рисунок 110***).

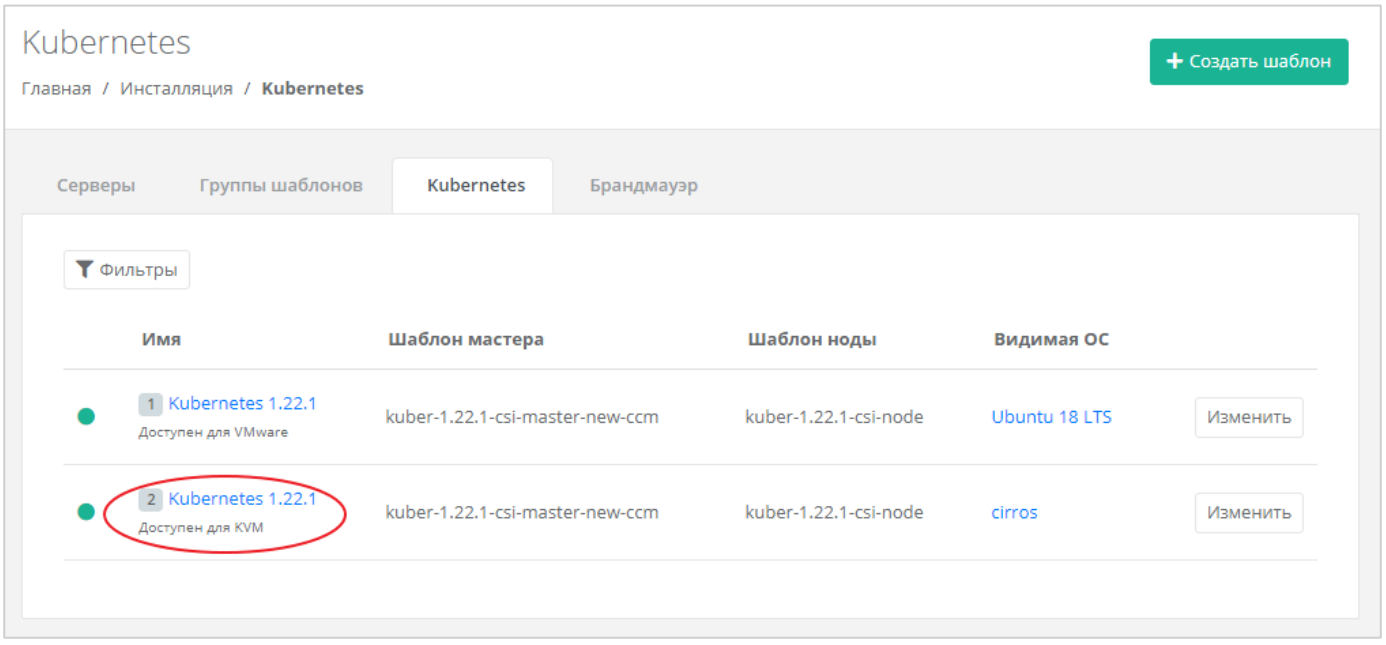

Рисунок 109

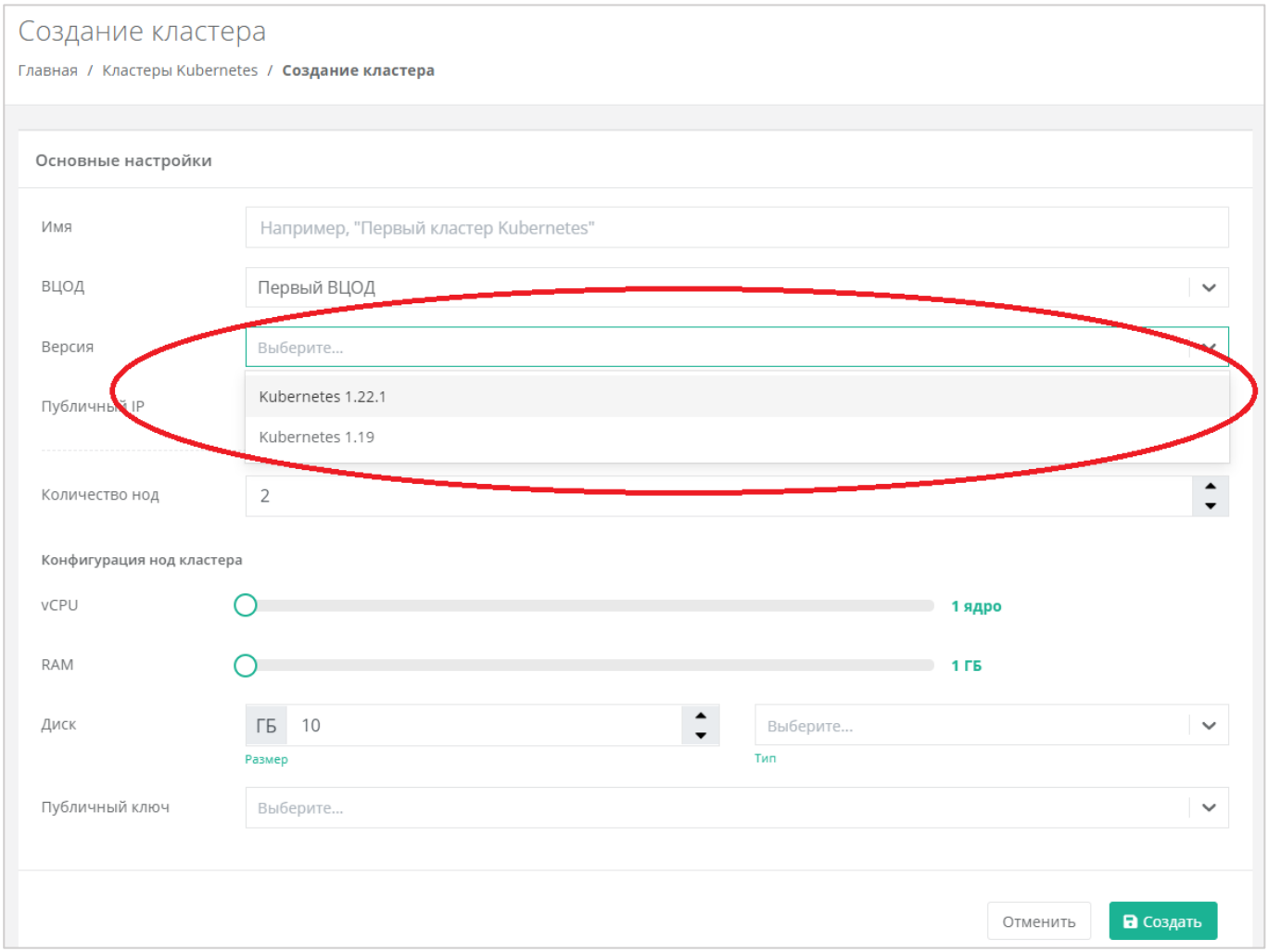

Рисунок 110

Также для последующего развёртывания кластеров в сегменте РУСТЭК/KVM необходимо произвести донастройку ресурсного пула.

Для этого в главном меню панели управления переходим в **Инсталляция** – **Ресурсы** – **Ресурсные пулы**. Выбираем ресурсный пул KVM **(*Рисунок 111*)**.

В открывшемся окне заполняем следующие параметры:

- Название management-сети, в которой работает ECУ – название маршрутизируемой сети из пункта **2.2** инструкции.
- Адрес ECУ в management-сети, по которому будет доступно API – адрес BM ESU-box в маршрутизируемой сети, выданный в пункте **2.2** (смотрим в панели РУСТЭК).
- Токен – токен пользователя (можно скопировать из настроек ресурсного пула vSphere).

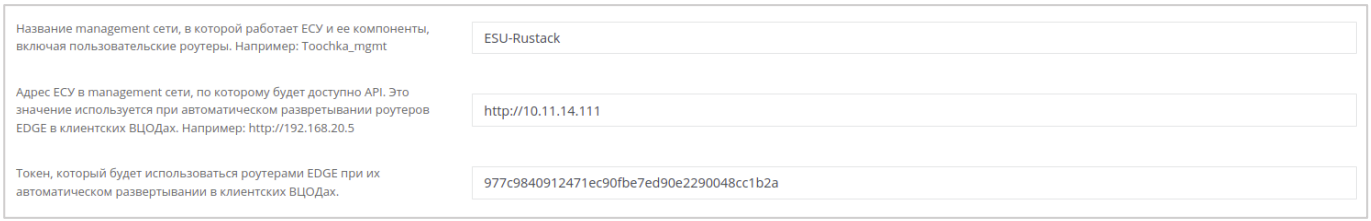

Рисунок 111

11.3. Создание кластеров Kubernetes в РУСТЭК-ЕСУ

После того, как шаблоны и ресурсный пул настроены, можно переходить к созданию кластеров Kubernetes.

Для этого в главном меню панели управления переходим в **Кластеры Kubernetes** и нажимаем «Создать кластер» (***Рисунок 112***).

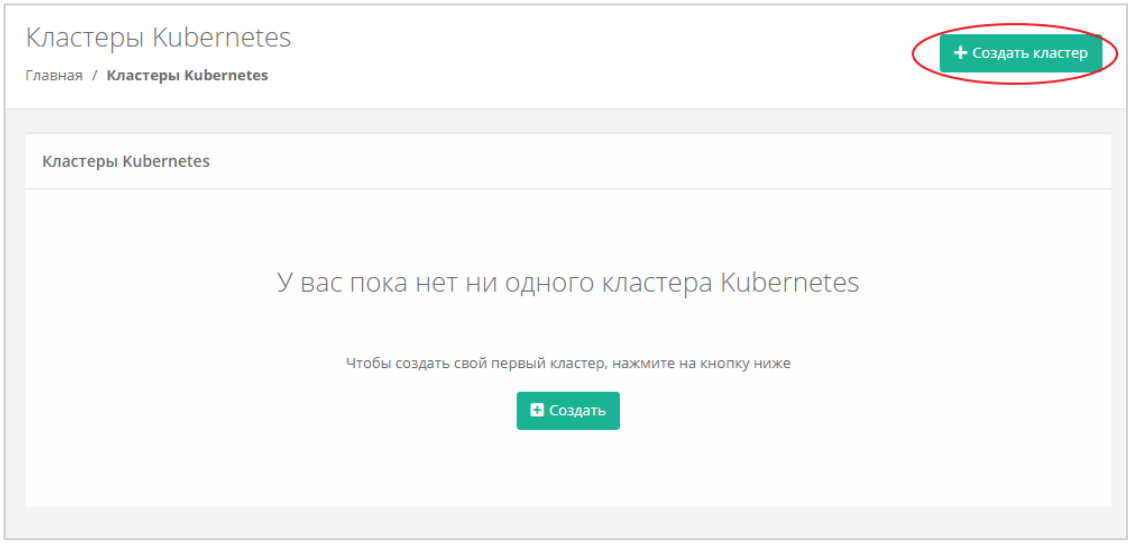

Рисунок 112

В открывшемся окне заполняем следующие параметры (***Рисунок 113***):

- Имя – произвольное наименование кластера.
- ВЦОД – выбор необходимого ВЦОД, либо создание нового.
- Версия – выбор версии Kubernetes.
- Публичный IP – выбор параметров публичного IP-адреса:
  - Отключен – кластер Kubernetes не будет иметь публичного IP-адреса.
  - Новый – получение нового IP-адреса из пула публичных адресов.
  - Случайный – использование выделенного для ВЦОД свободного IP-адреса, в случае отсутствия такого – получение нового из пула публичных адресов.
- Количество нод – выбор количества нод для кластера.
- Конфигурация нод кластера – выбор параметров конфигурации нод:
  - CPU.
  - RAM.
  - Диск:
    - Размер диска.
    - Тип диска (SSD, SAS, SATA).
- Публичный ключ – выбор публичного ключа и возможность создания нового.

Все поля должны быть заполнены. Также необходимо добавить публичный ключ (его можно сгенерировать в панели управления) он нужен для доступа мастер ноды к остальным нодам кластера.

Когда все поля заполнены нажимаем «Создать».

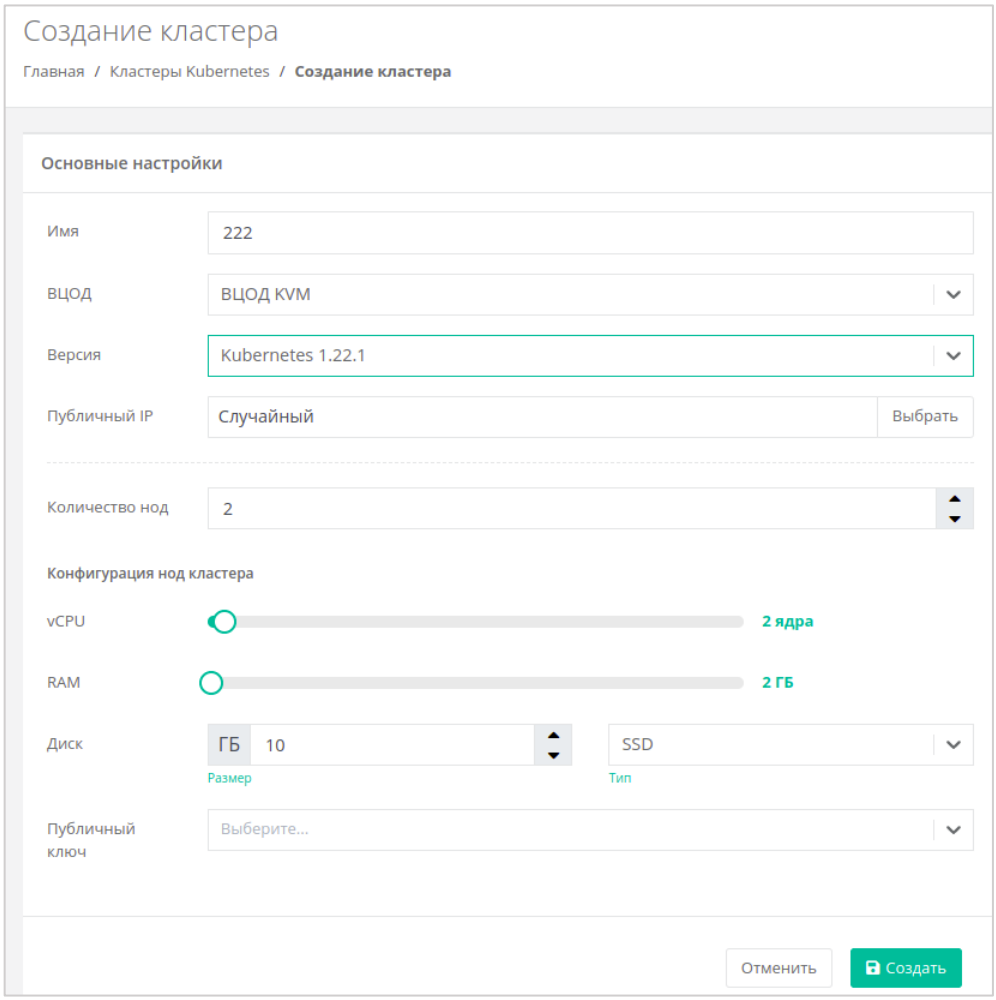

Рисунок 113

После создания кластер отобразится в панели управления (***Рисунок 114***).

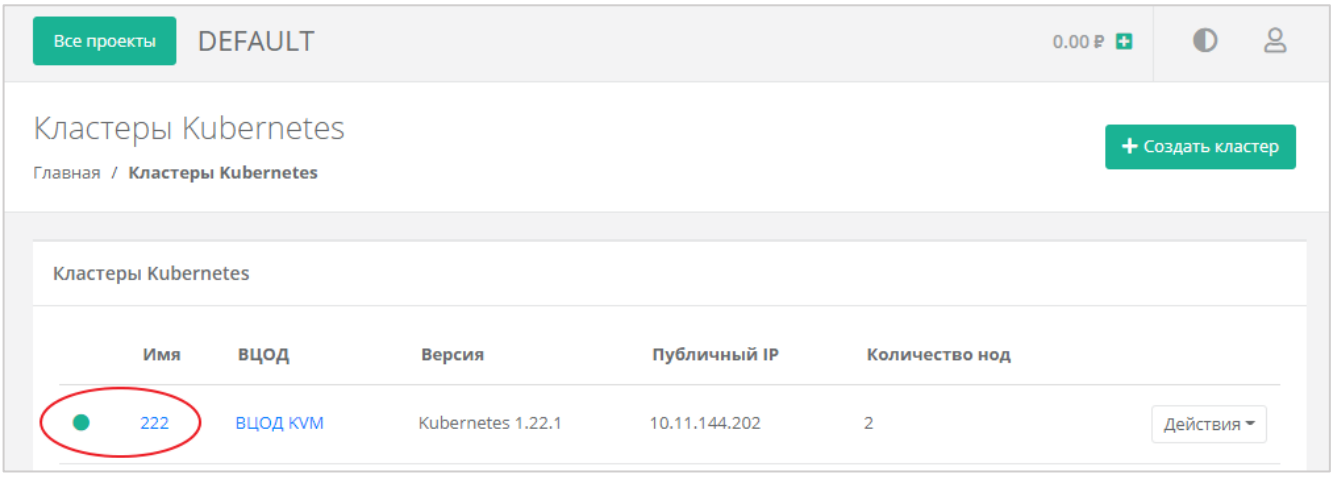

Рисунок 114

Ноды кластера также можно увидеть в меню **Облачные вычисления** – **ВЦОД** – **Серверы** и управлять ими как обычными серверами – изменять конфигурацию и управлять состоянием сервера (***Рисунок 115***).

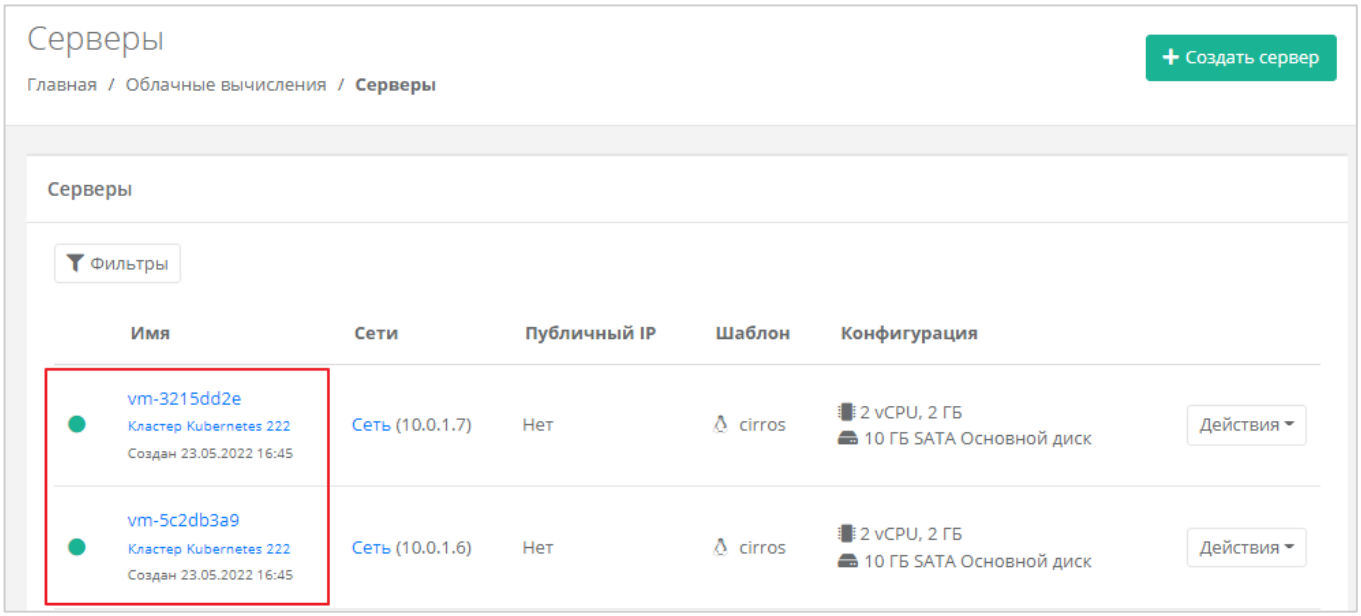

Рисунок 115

11.4. Особенности и поддерживаемый функционал

Особенности:

- Кластер развёртывается только в сервисной сети ВЦОДа (созданной автоматически при создании ВЦОД).
- Требуется наличие пользовательского публичного ключа в профиле, так как ноды будут создаваться без пароля, но с ключом. Это упрощает процедуру развертывания и настройку опций развёртывания для пользователя.
- Сервисы k8s, отвечающие за работоспособность кластера, физически запущены на одной ВМ. В случае ее «падения» кластер будет неуправляем до момента ее включения.
- Мастер-нода недоступна для управления пользователем и располагается в management-сети.

Поддерживаемый функционал:

- Балансировщики нагрузки в кластере Kubernetes (доступны только для сегмента VMware vSphere).
- Создание Persistence Volume Claims (доступны в обоих сегментах, но только создание – изменение недоступно).

12. Расширенная настройка

12.1. Настройка NGINX реверс-прокси

РУСТЭК-ЕСУ должна работать с конечными пользователями только по https.

Рекомендуется настроить проксирование РУСТЭК-ЕСУ для конечных пользователей на специально организованном реверс-прокси, например, nginx. Для упрощения построения проксирования в РУСТЭК-ЕСУ открыт порт 80.

Ниже приведён пример минимальной конфигурации файла nginx, который необходимо создать **/etc/nginx/conf.d/<любое имя>.conf**, где:

- **<your_domain>** – доменное имя сервера nginx.
- **<ip_esu-box>** – IP адрес по которому доступна панель управления.
- **<path_to_crt>** –- путь к SSL-сертификату.
- **<path_to_key>** – путь к ключу.

```
server {
        server_name <your_domain>;

        location / {
                proxy_read_timeout      1800;
                proxy_connect_timeout   1800;
                proxy_redirect          off;

                proxy_set_header        Host                $http_host;
                proxy_set_header        X-Real-IP           $remote_addr;
                proxy_set_header        X-Forwarded-For     $proxy_add_x_forwarded_for;
                proxy_set_header        X-Forwarded-Proto    $scheme;
                proxy_set_header        X-Frame-Options     SAMEORIGIN;

                proxy_set_header        Upgrade             $http_upgrade;
                proxy_set_header        Connection           "upgrade";

                proxy_pass      http://<ip_esu-box>:80;
                proxy_buffering off;

        }

    listen 443 ssl;
    client_max_body_size 150G;
    proxy_ssl_session_reuse off;
    ssl_certificate <path_to_crt>/fullchain.pem;
    ssl_certificate_key <path_to_key>/<your domain>/privkey.pem;
}
```

После создания файла конфигурации необходимо запустить службу nginx, для этого выполним команду:

```
systemctl start nginx
```

Затем необходимо добавить службу nginx в автозапуск, для этого выполним команду:

```
systemctl enable nginx
```

Документация по настройке nginx: https://nginx.org/ru/docs/

Примечания:

- не следует работать с РУСТЭК-ЕСУ напрямую по порту 80, так как в этом случае не будет работать часть функционала, связанного с асинхронными обновлениями данных в браузере пользователя;

- по соображениям безопасности 80-й порт может быть отключён в будущих релизах;
- обратите внимание, что кэширование на стороне реверс-прокси отключено. Замечено, что при использовании модуля modsecurity кэширование на стороне nginx может непреднамеренно включиться.

12.2. Настройка управления DNS-зонами в РУСТЭК-ЕСУ

РУСТЭК-ЕСУ имеет службу, позволяющую пользователям управлять ресурсными записями делегированных в неё доменов. Зоны, как водится, должны раздаваться как минимум с двух серверов, например, с пакетом BIND, работающих и настроенных отдельно от РУСТЭК-ЕСУ, но находящихся в той же сети. Раннер в РУСТЭК-ЕСУ выполняет роль так называемого каталога зон.

Обратите внимание, что нужна сетевая связность не только от BIND к РУСТЭК-ЕСУ, но и в обратную сторону.

Для примера в инсталляции РУСТЭК были развёрнуты два сервера на базе Ubuntu 18.04 LTS в той же сети, что и ESU-box. Также для наших серверов и ESU-box необходимо добавить дополнительные правила брандмауэра в созданный раннее профиль безопасности (***Рисунок 116***).

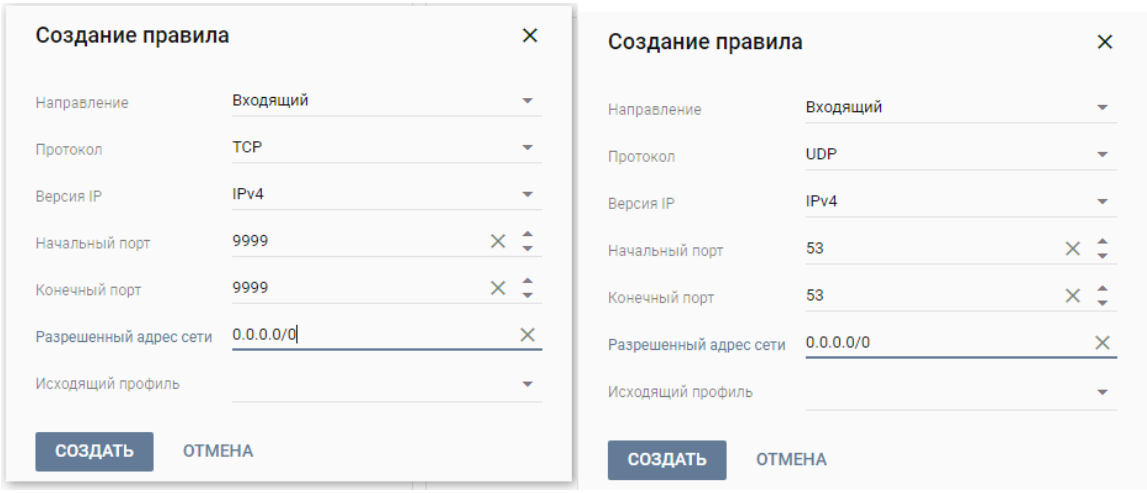

Рисунок 116

Ниже показан пример конфигурации BIND 9.11 для работы с каталогом зон из РУСТЭК-ЕСУ.

Пример конфигурации приведен на базе BIND из Ubuntu 18.04 LTS.

Устанавливаем:

```
apt-get install -y bind9 bind9utils bind9-doc
```

Устанавливаем хостнейм на наши серверы командой:

```
hostnamectl set-hostname <name>
```

Представим, что ESU-box расположена по адресу 10.11.14.111. Тогда конфигурационный файл **/etc/bind/named.conf.options** должен выглядеть так:

```
options {
    directory "/var/cache/bind/";

    allow-transfer { none;};
    dnssec-validation no;
    minimal-responses yes;

    auth-nxdomain no;
    listen-on port 53 { any; };

    recursion no;
    catalog-zones {
        zone "catalog.local" default-masters {
            10.11.14.111 port 9999;
        };
    };

    allow-notify {
        10.11.14.111;
    };
};

zone "catalog.local" {
    type slave;
    file "catalog.db";
    masters { 10.11.14.111 port 9999; };
};
```

Запускаем службу командой:

```
systemctl start bind9
```

Добавляем в автозапуск службу BIND:

```
systemctl enable bind9
```

Для созданных серверов необходимо добавить DNS записи (имена).

Для данного примера это было сделано с помощью редактирования файла **/etc/hosts** на ВМ ESU-box.

После произведённой настройки имена DNS-серверов, а также e-mail администратора, следует указать в самой РУСТЭК-ЕСУ на уровне провайдера в меню **Администрирование** – **Партнёры** (***Рисунок 117***).

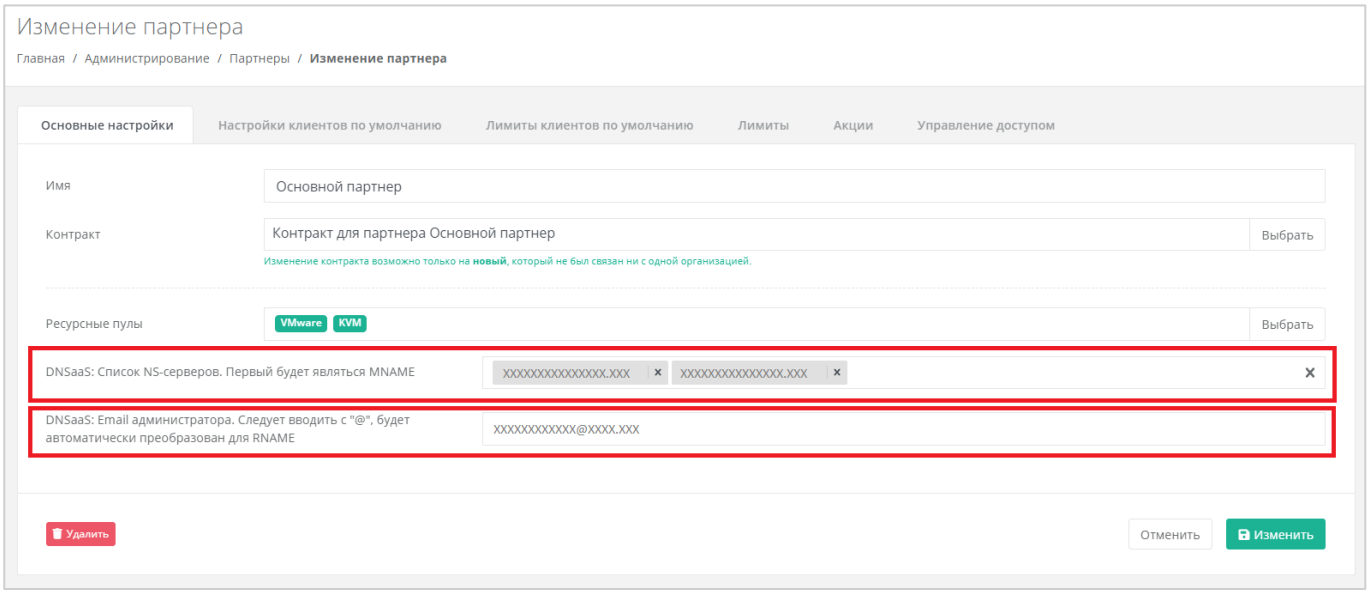

Рисунок 117

После успешной настройки в главном меню панели РУСТЭК-ЕСУ появится пункт **Доменные зоны**, из которого можно управлять доменными зонами и записями в них (***Рисунок 118***).

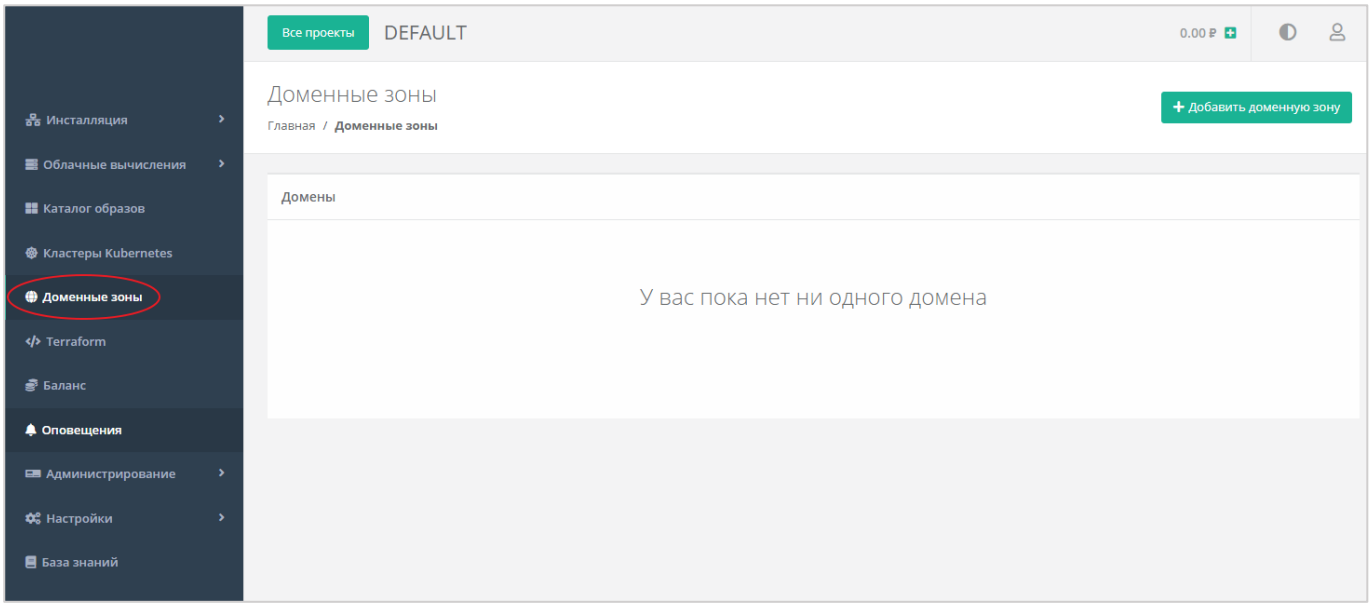

Рисунок 118

12.3. Настройка сети для роутеров (edge) сегмента VMware vSphere

Базовая установка РУСТЭК-ЕСУ размещает пользовательские роутеры сегмента VMware в своей сервисной сети. Это удобно для быстрого запуска, но может вызывать проблемы при большом числе клиентов (размер сервисной сети ограничит количество клиентов сегмента VMware).

В таком случае необходимо создать отдельную сеть для роутеров внутри РУСТЭК, например, Edge_network (***Рисунок 119***).

Для этого в панели РУСТЭК необходимо перейти в раздел **Сеть** – **Сети** и нажать «Создать».

- Имя – указывается произвольное.

- Тип сегментации – VLAN.
- Номер VLAN – номер выделенного VLAN для внешней сети Единой системы управления.
- Внешняя – снять чек-бокс.
- Безопасность портов – указывается опционально. Данный функционал добавляет возможность использовать фаерволл на уровне порта виртуальной машины средствами ПАК.

  Нажать «Создать».

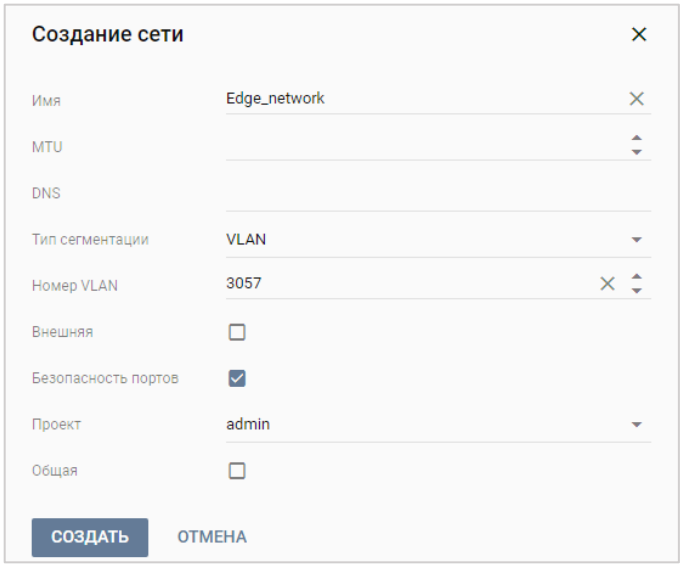

Рисунок 119

Далее необходимо создать подсеть для созданной сети (***Рисунок 120***).

Для этого перейдите в раздел **Сети** – **Подсеть** и нажмите «Создать», далее необходимо заполнить поля:

- Имя – указывается произвольное.
- Сеть – выбрать сеть, созданную на предыдущем этапе.
- Версия протокола – Ipv4.
- Адрес сети – указать cidr.
- Шлюз – указать шлюз.
- DHCP – снять чек-бокс.
- DNS-серверы – прописать по желанию.

Нажать «Создать».

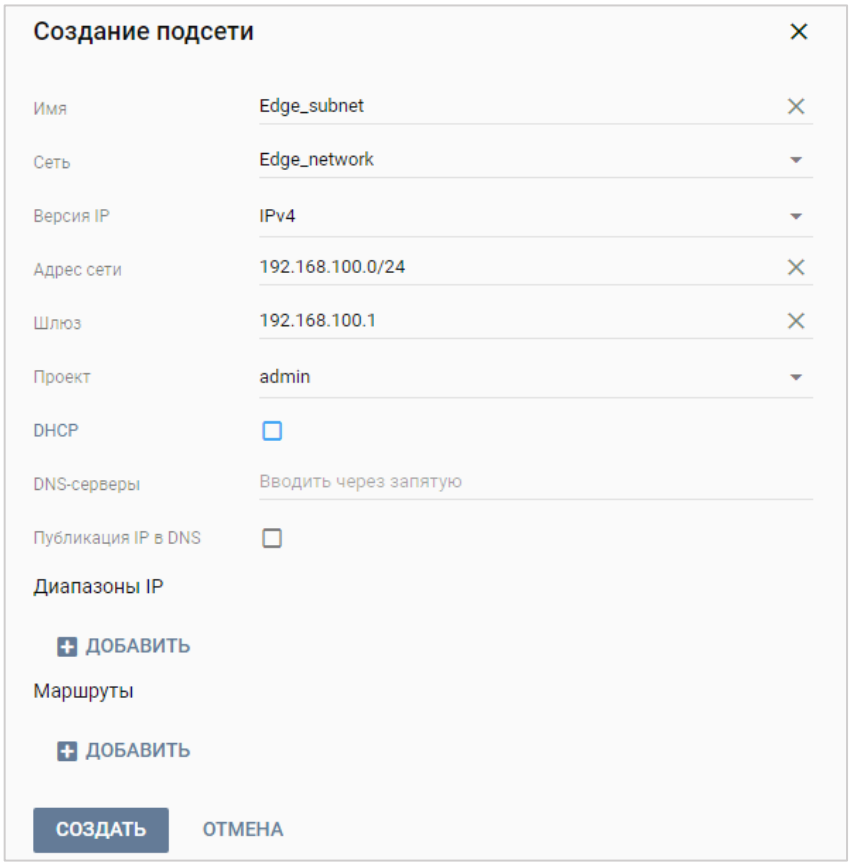

Рисунок 120

Затем необходимо подключить ESU-box (сервер с РУСТЭК-ЕСУ) к этой сети.

Для этого перейдём в раздел **Серверы**, выберем сервер с установленной РУСТЭК-ЕСУ (ESU-box), правой кнопкой мыши раскроем меню действий и выберем «Сети», затем добавим новую созданную сеть (***Рисунок 121***, ***Рисунок 122***).

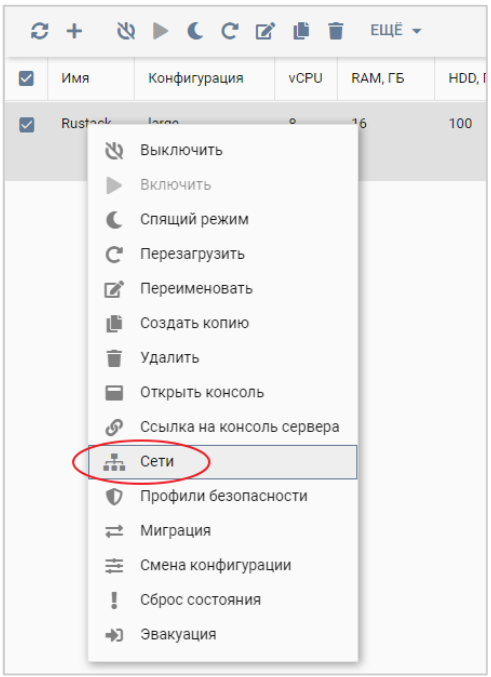

Рисунок 121

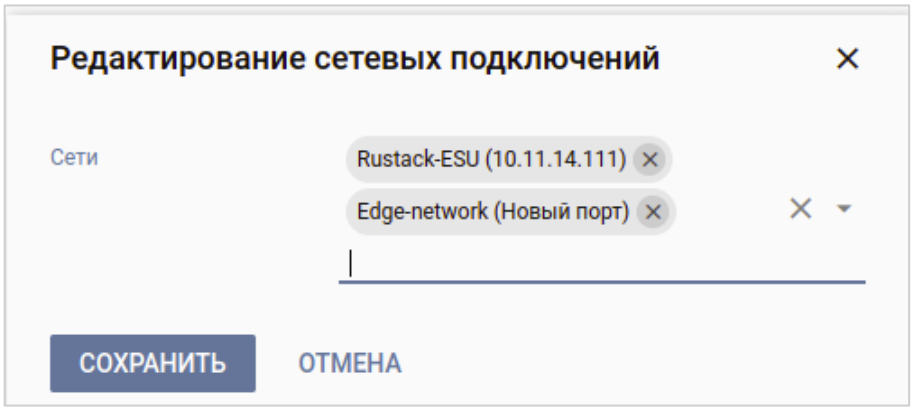

Рисунок 122

Теперь узнаём IP –адрес, назначенный для ESU-box в сети Edge_network, для этого обновим страницу в меню **Серверы** (***Рисунок 123***):

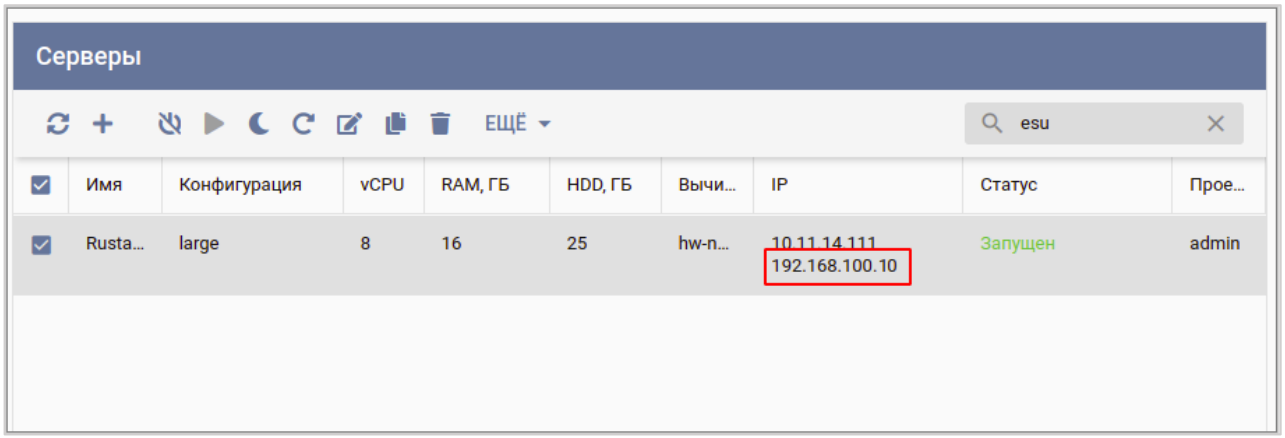

Рисунок 123

Затем подключаемся по ssh к ESU-box, где необходимо настроить наш новый сетевой интерфейс.

Сначала необходимо узнать имя нового сетевого интерфейса для этого выполняем команду:

```
ip a | grep en
```

В нашем случае имя нового сетевого интерфейса enp7s0

Затем настраиваем этот интерфейс, для этого выполняем следующие команды:

```
sudo nano /etc/network/interfaces
```

В содержимое файла вставить:

```
auto enp7s0
iface enp7s0  inet static
 address 192.168.100.10
 netmask 255.255.255.0
 gateway 192.168.100.1
```

Затем необходимо настроить DHCP-сервер на ESU-box, для нового сетевого интерфейса. Для этого выполним следующие команды:

Добавляем имя нового интерфейса в файл **/etc/default/isc-dhcp-server** (***Рисунок 124*****.**

```
sudo vi /etc/default/isc-dhcp-server
```

```
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4=""
INTERFACESv6=""
# BEGIN ANSIBLE MANAGED BLOCK
INTERFACESv4="ens160 enp7s0"
# END ANSIBLE MANAGED BLOCK
```

Рисунок 124

Теперь производим настройку DHCP-сервера (***Рисунок 125***):

```
sudo vi /etc/dhcp/dhcpd.conf
```

В содержимое файла вставить:

```
subnet 192.168.100.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option routers 192.168.100.1;
  option domain-name-servers 8.8.8.8;
  range 192.168.100.10 192.168.100.255;
  default-lease-time 600;
  max-lease-time 10800;
}
```

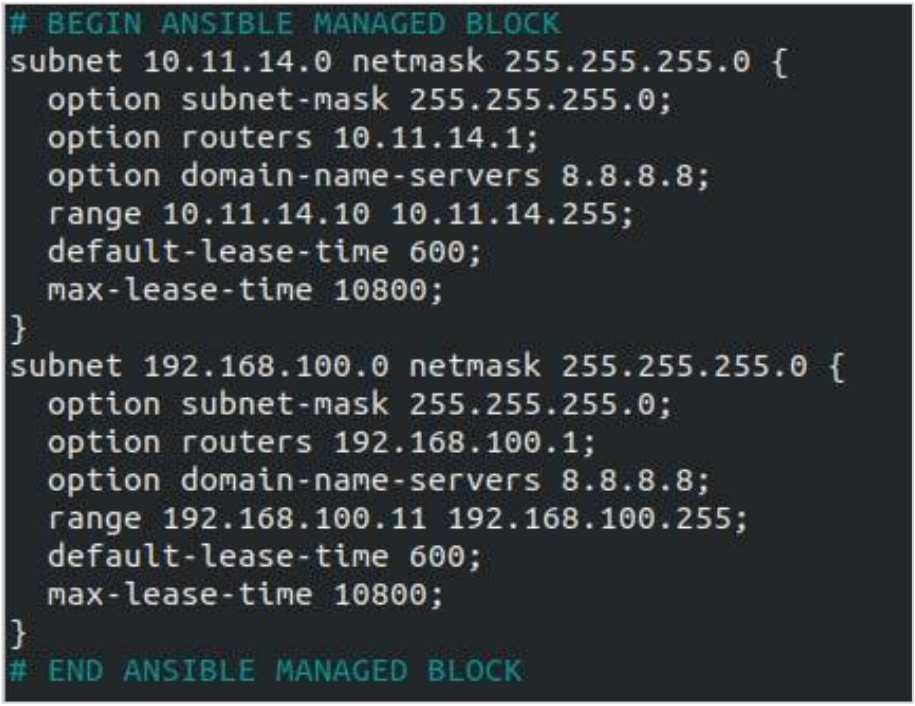

Рисунок 125

Перезагружаем службы DHCP-сервера и сети:

```
sudo service isc-dhcp-server restart
sudo service networking restart
```

После этого необходимо создать и настроить сеть (portgroup на dvswitch) в VMware vSphere (***Рисунок 126***, ***Рисунок 127***).

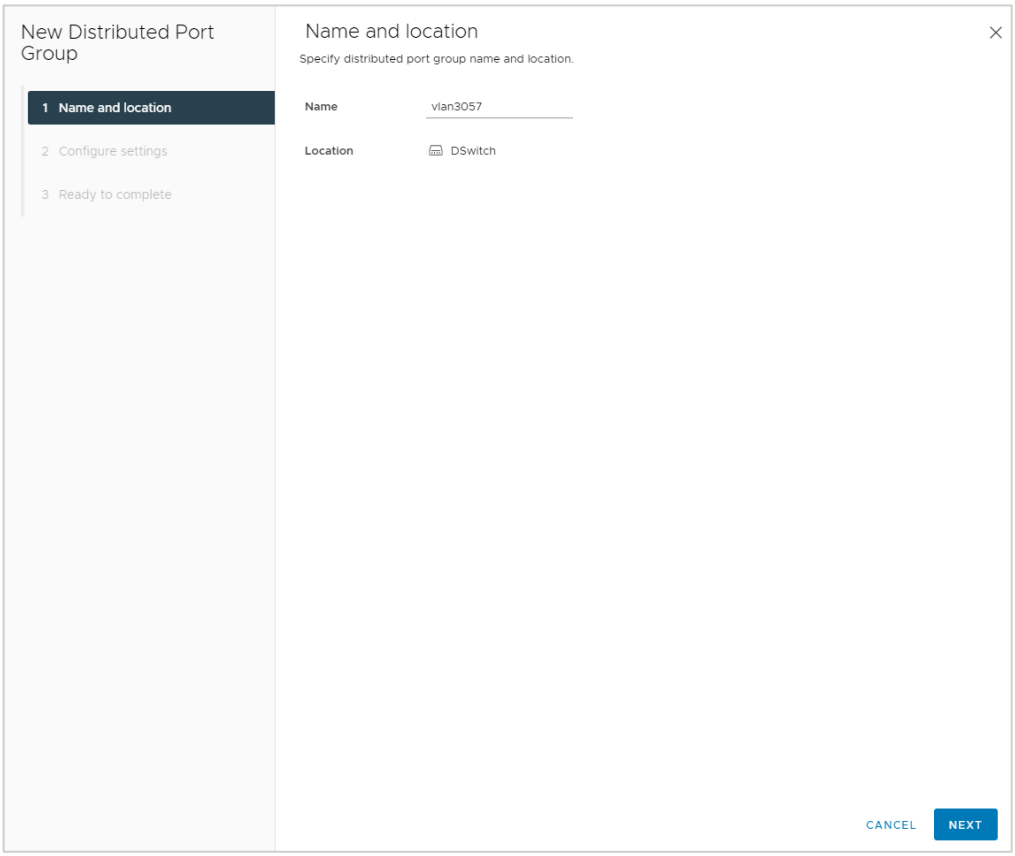

Рисунок 126

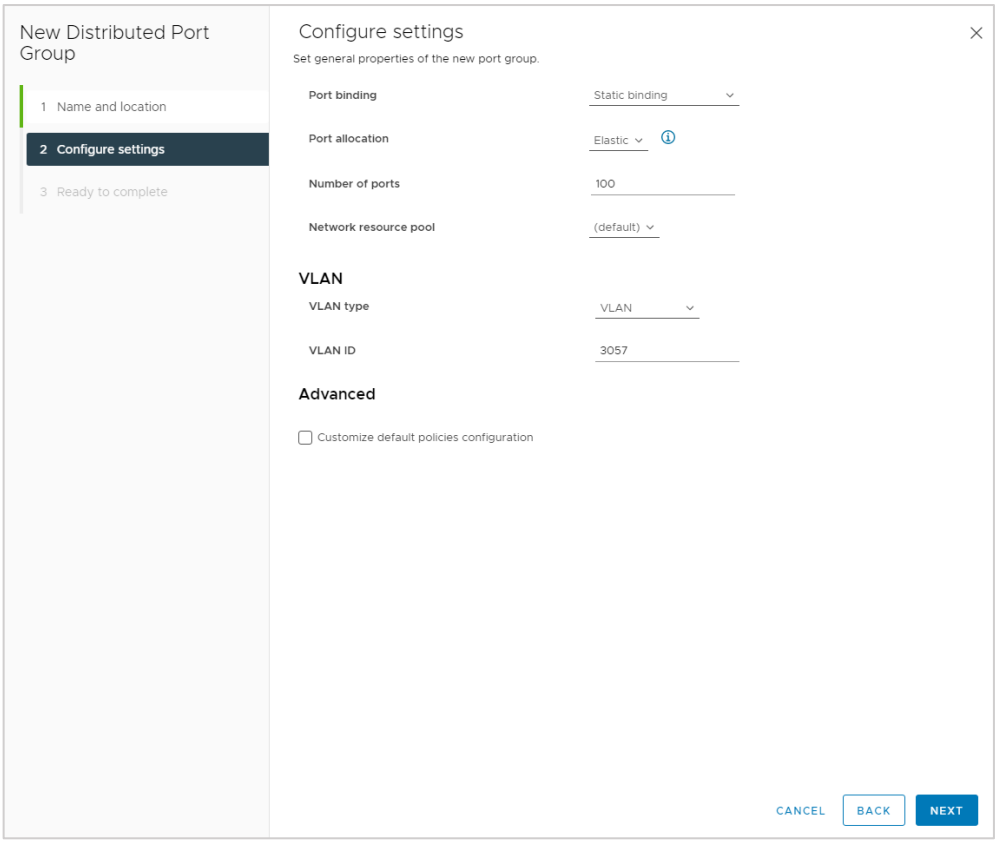

Рисунок 127

Далее необходимо указать через web-интерфейс в настройках ресурсного пула VMware данную сеть как management-сеть для роутеров.

Для этого в панели управления РУСТЭК-ЕСУ переходим в меню **Инсталляция** – **Ресурсы** – **Ресурсные пулы**.

Выберем ресурсный пул VMware vSphere и изменим следующие настройки (***Рисунок 128***):

- Название management-сети для пользовательских роутеров – укажем название нашей сети в VMware vSphere.
- Адрес ЕСУ в management-сети, в которой будут создаваться роутеры – указываем адрес сервера ESU-box в новой сети (раннее смотрели его в панели РУСТЭК).

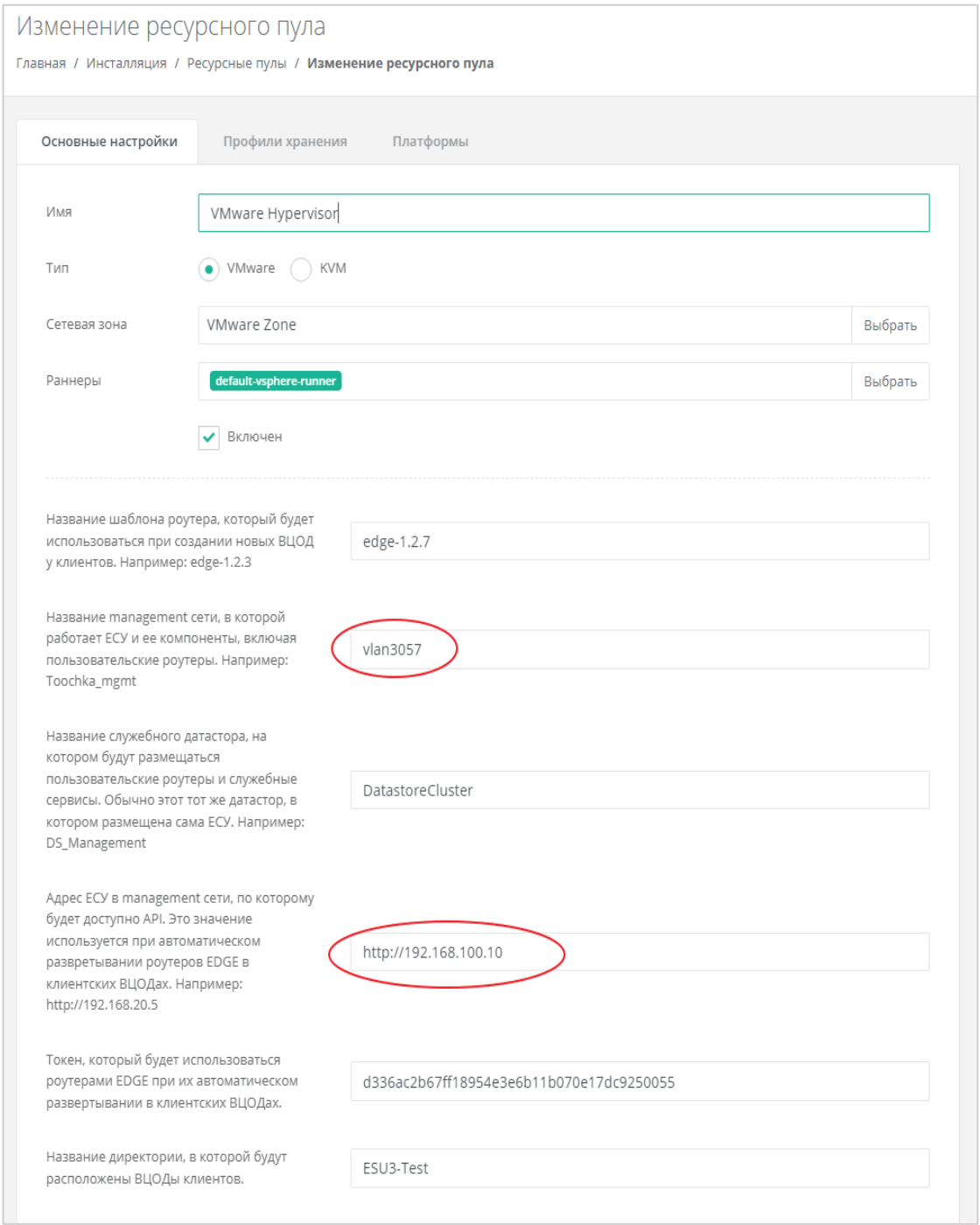

Рисунок 128

На этом настройка завершена.

Следует отметить, что уже созданные Роутеры (edge) останутся в той сети, в которой были созданы. Новые же будут создаваться в новой настроенной сети.

Проверим это, создав новый ВЦОД в сегменте VMware vSphere (***Рисунок 129***).

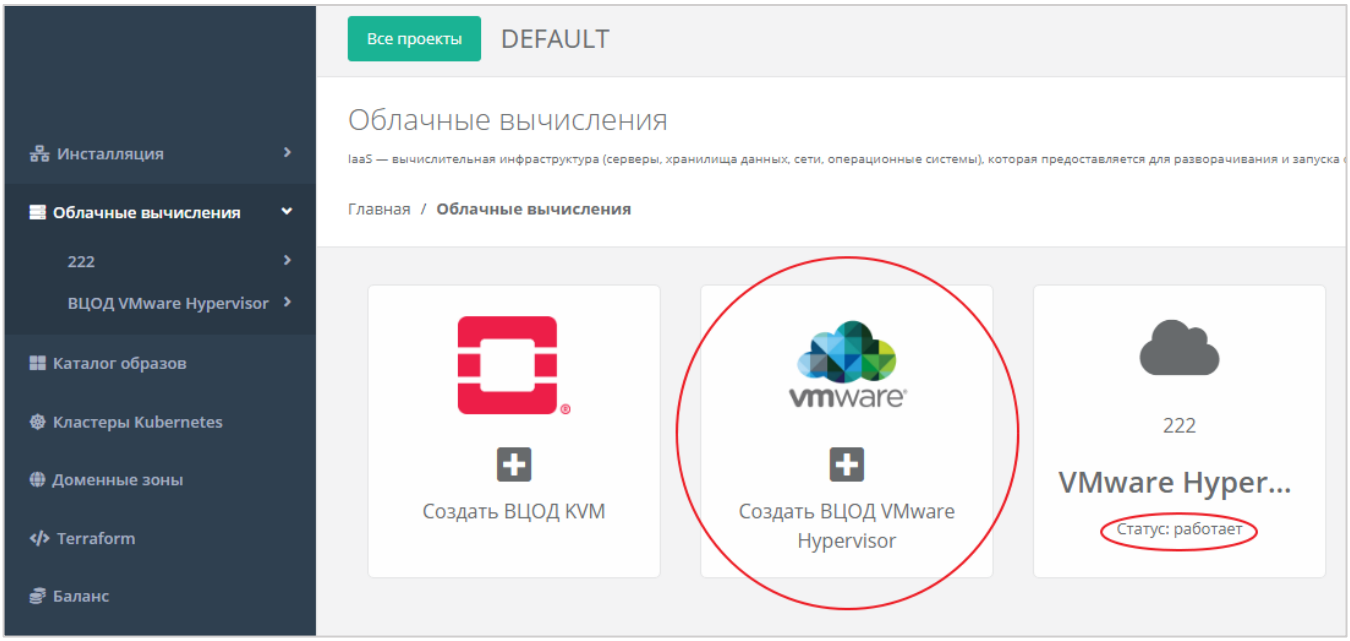

Рисунок 129

После создания ВЦОД перейдём в панель VMware vSphere и убедимся, что роутер (edge), созданный внутри нового ВЦОД, подключен к новой настроенной сети (***Рисунок 130***).

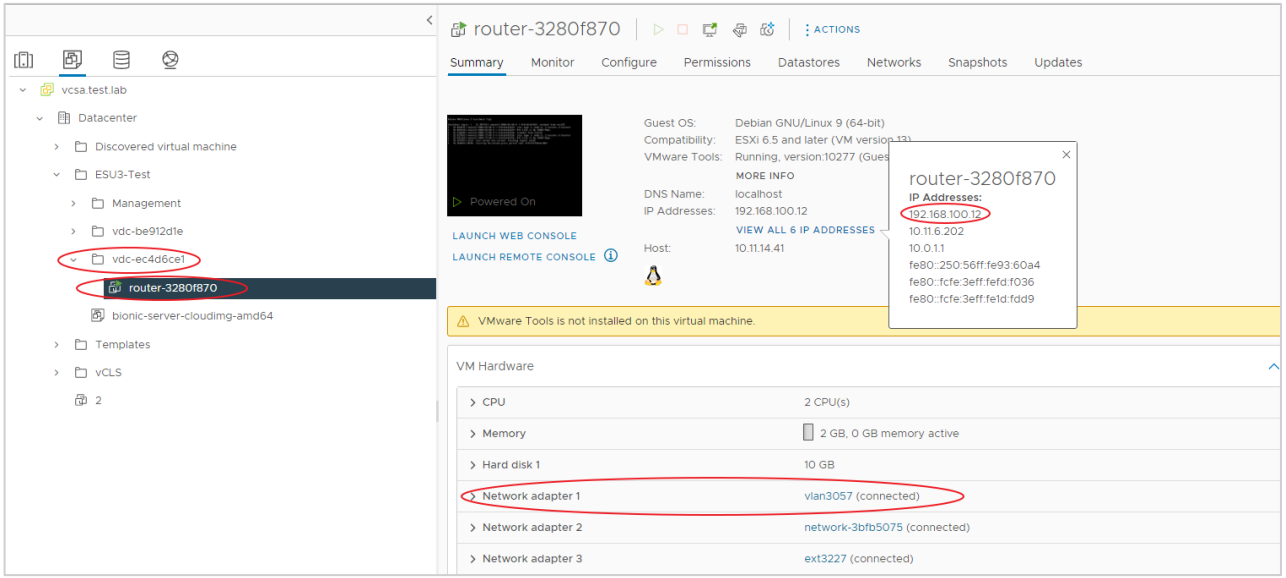

Рисунок 130

12.4. Универсальный скрипт развёртывания

Скрипт развёртывания используется в процедуре создания шаблонов для последующего развёртывания серверов в панели управления РУСТЭК-ЕСУ. Создать шаблоны можно в меню **Инсталляция** – **Шаблоны** – **Серверы**.

Для начала необходимо подготовить шаблон и загрузить на платформы виртуализации согласно инструкциям раздела **6** (для сегмента РУСТЭК/KVM) и из раздела **9** (для сегмента VMware vSphere).

Сам скрипт пишется на языке JavaScript и должен содержать функцию getMetadata(vmInfo, userData), возвращающую набор полей для передачи через EC2.

Вам понадобится добавить в меню **Инсталляция** – **Шаблоны** – **Сервер**ы к шаблонам ВМ следующие поля во вкладке «Поля для скрипта» при заведении шаблона (***Рисунок 131***):

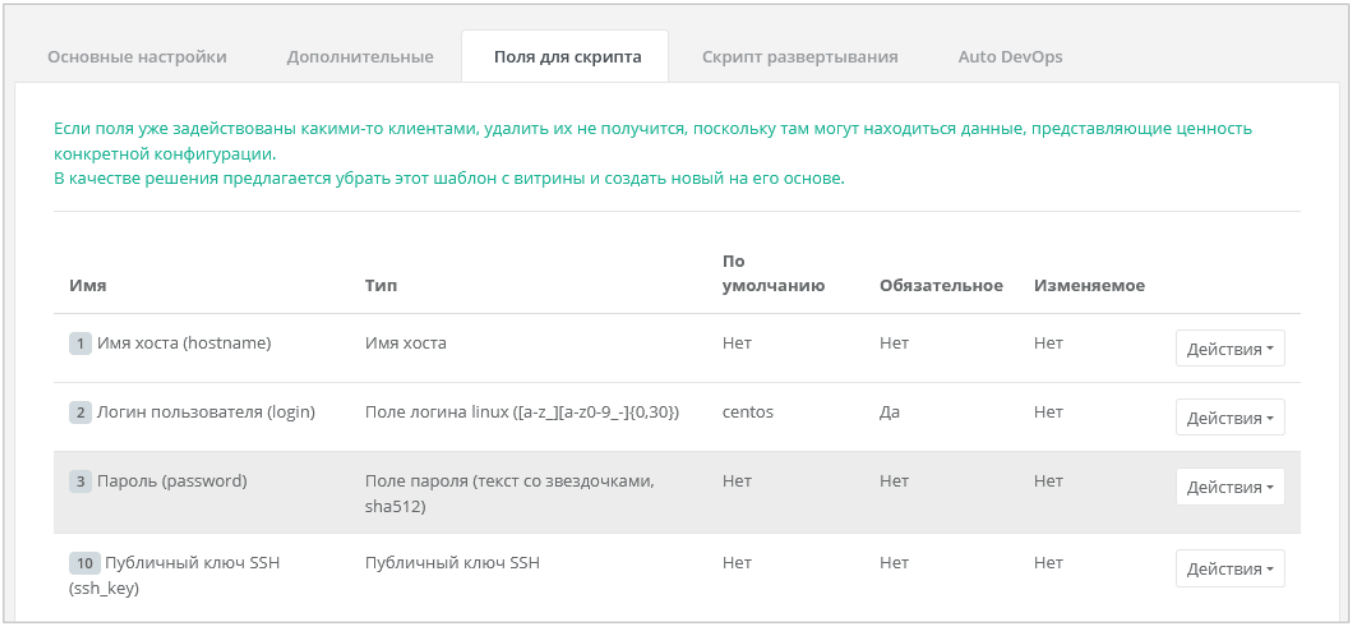

Рисунок 131

Универсальный скрипт, подходящий для Ubuntu 16, Ubuntu 18, Ubuntu 20, Debian 9, Debian 10, Centos 7, Centos 8:

```
from loguru import logger
from rest_framework import serializers

"""
ESU metadata script
Version 3.1 (2021-07-02)

CUSTOM!
"""



def get_metadata(vm, user_data):
    # В логи контейнера API попадет следующая информация:
    logger.info('Create metadata for {}. vm: {}, user_data: {}', vm.template, vm, user_data)

    # В отличии от user_data['hostname'], в vm.hostname всегда что-то есть. Если не от пользователя,
    # то от системы:
    hostname = vm.hostname

    # Фрагменты для подмешивания в YAML cloud-config'а
    ssh_fragment = password_fragment = ''

    # Если пользователь указал ключ, добавим его
    if user_data['ssh_key']:
        ssh_fragment = fr"""
    ssh_authorized_keys:
      - "{user_data['ssh_key']}"
"""

    # Если пользователь указал пароль, добавим его
    if user_data['password']:
        password_fragment = fr"""
    passwd: "{user_data['password']}"
    lock_passwd: false
"""

    # Если пользователь не указал ни ключ, ни пароль, покажем ошибку
    if not ssh_fragment and not password_fragment:
```

```
        raise serializers.ValidationError('Чтобы иметь доступ на сервер, необходимо или ввести пароль или выбрать публичный ключ. Допустимо также задать пароль вместе с публичным ключом.')


    cloud_config = fr"""
#cloud-config
debug:
  verbose: false
cloud_init_modules:
  - migrator
  - seed_random
  - bootcmd
  - write-files
  - growpart
  - resizefs
  - set_hostname
  - update_hostname
  - update_etc_hosts
  - users-groups
  - ssh
bootcmd:
  - [ cloud-init-per, once, rmdefaultuser1, userdel, -r, centos ]
  - [ cloud-init-per, once, rmdefaultuser2, userdel, -r, debian ]
  - [ cloud-init-per, once, rmdefaultuser3, userdel, -r, ubuntu ]
  - [ sh, -c, echo "your_OS ver.1.10" ]
users:
  - name: {user_data['login']}
    groups: [adm, audio, cdrom, dialout, dip, floppy, lxd, netdev, plugdev, sudo, video]
    sudo: ["ALL=(ALL) NOPASSWD:ALL"]
    shell: /bin/bash
{password_fragment}
{ssh_fragment}
disable_root: true
timezone: "Europe/Moscow"
package_update: false
manage_etc_hosts: localhost
fqdn: "{hostname}"
datasource:
  Ec2:
    strict_id: false
```

```
    timeout: 5
    max_wait: 5
    metadata_urls:
      - http://169.254.169.254:80
"""


    # Возвращаем данные для сервера метадаты
    return {
        'user_data': cloud_config,
        'hostname': hostname,
        'instance-id': vm.short_id
    }
```

12.5. Подготовка сервера с Veeam Backup&Replication для работы с РУСТЭК-ЕСУ

Примечание: Перед настройкой Veeam Backup&Replication необходимо подготовить хранилище для резервных копий.

1. Разворачиваем базовую ОС Windows согласно техническим требованиям продукта Veeam.
2. Устанавливаем Veeam Backup&Replication 11 *(с другими версиями РУСТЭК-ЕСУ не работает).*
3. Настраиваем взаимодействие Veeam Backup&Replication и VMware vSphere.
4. Настраиваем ScaleOut Repository.
5. Устанавливаем и настраиваем OpenSSH внутри OS Windows.
6. Настраиваем Veeam Backup&Replication-раннер в панели управления РУСТЭК-ЕСУ.

Пункты 1 – 3 выполняем согласно официальной документации:
https://helpcenter.veeam.com/docs/backup/vsphere/distributed.html?ver=110

Пункт 4 выполняем согласно документации:
https://helpcenter.veeam.com/docs/backup/vsphere/backup_repository_sobr.html?ver=110

РУСТЭК-ЕСУ взаимодействует с Veeam Backup&Replication отправкой команд через PowerShell. Для этого на сервере, где доступна оснастка Veeam Backup&Replication, должен стоять SSH-сервер.

Порядок настройки SSH-сервера:

- Скачать OpenSSH-Win64.zip отсюда https://github.com/PowerShell/Win32-OpenSSH/releases
- Разархивировать в C:\Program Files\OpenSSH-Win64
- Перейти в панель управления / Система / Advanced System Settings / Advanced / Environmental Variables (***Рисунок 132***):

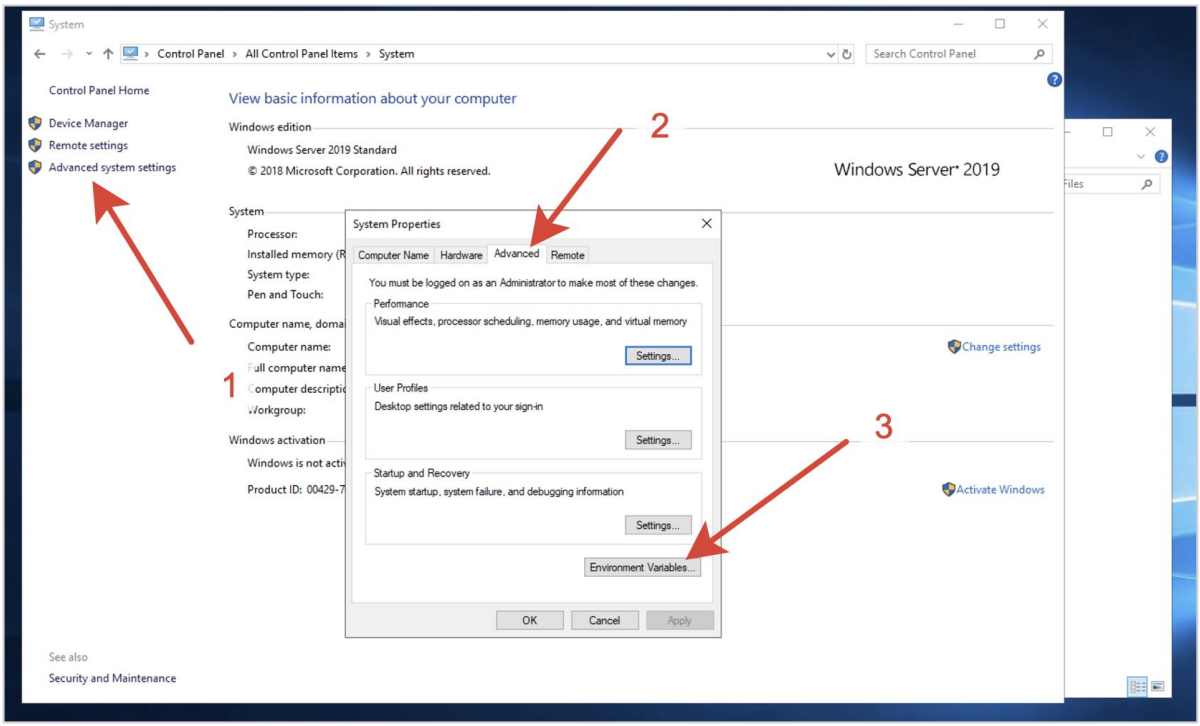

Рисунок 132

- В system variables (второй блок) выбрать path, нажать редактировать (***Рисунок 133***):

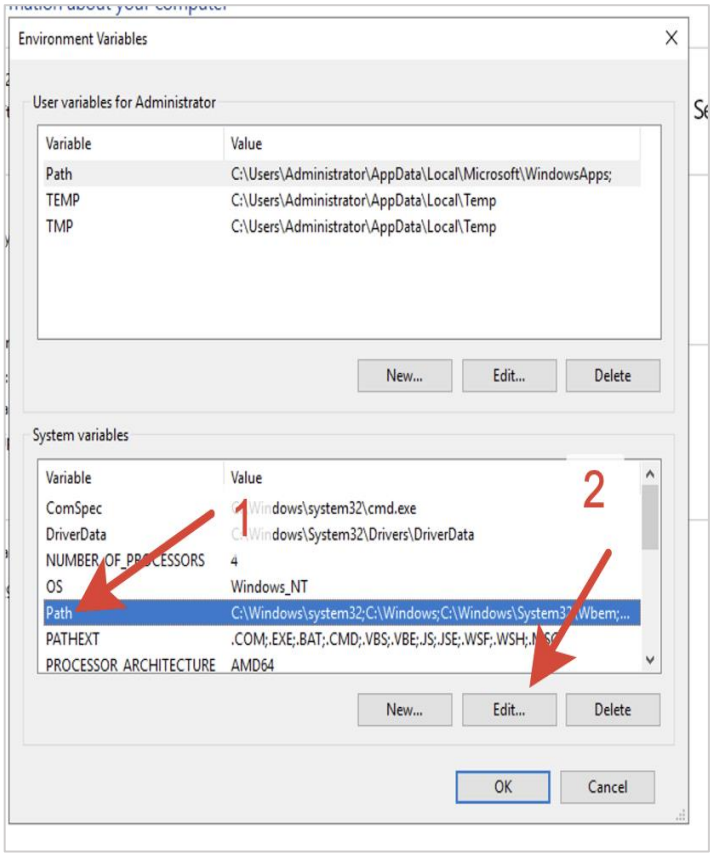

Рисунок 133

- Добавить туда C:\Program Files\OpenSSH-Win64 (***Рисунок 134***):

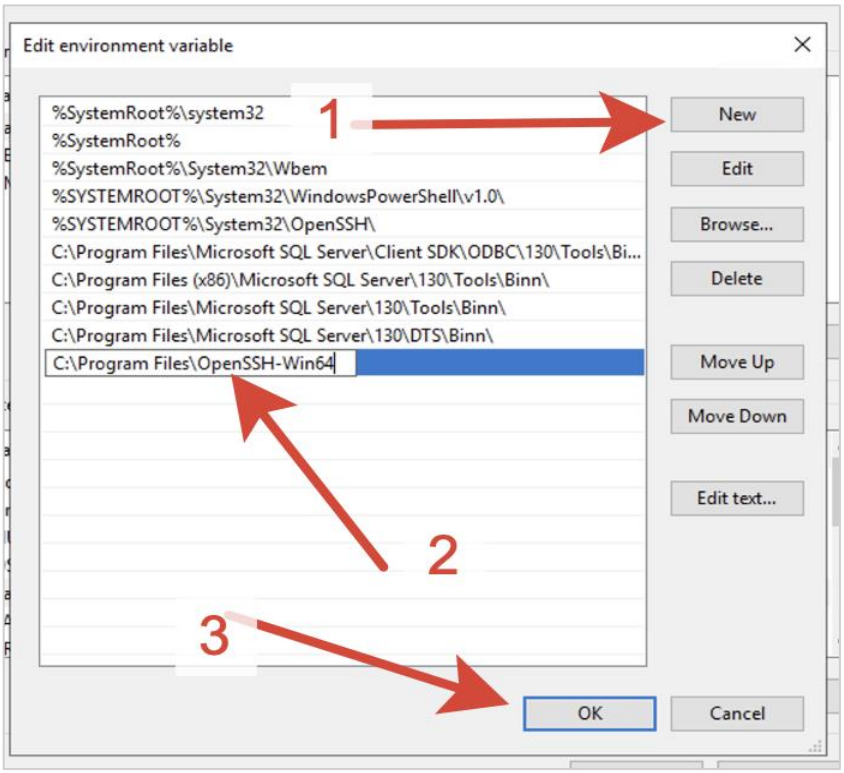

Рисунок 134

- Запустить powershell как администратор (***Рисунок 135***):

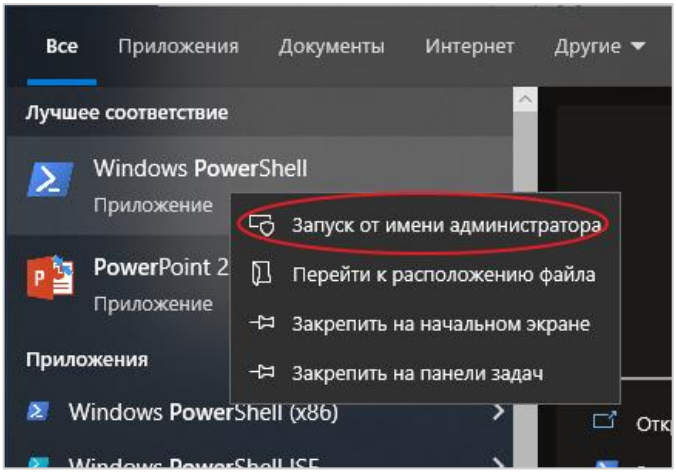

Рисунок 135

- Перейти в C:\Program Files\OpenSSH-Win64
- Запустить .\install-sshd.ps1
- Если надпись "sshd and ssh-agent services successfully installed" появилась – все верно (***Рисунок 136***):

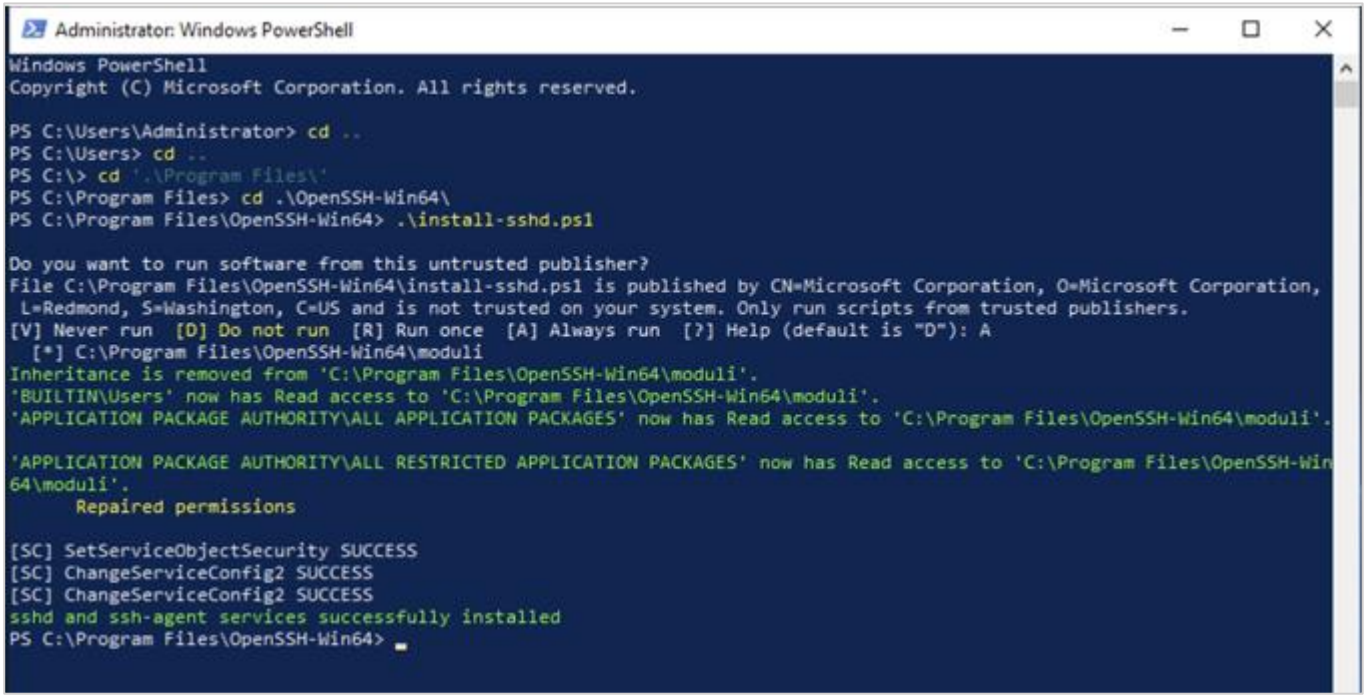

Рисунок 136

- Сгенерировать ключ хоста: .\ssh-keygen.exe -A
- Зайти в сервисы, включить и настроить автозапуск сервису OpenSSH (***Рисунок 137***):

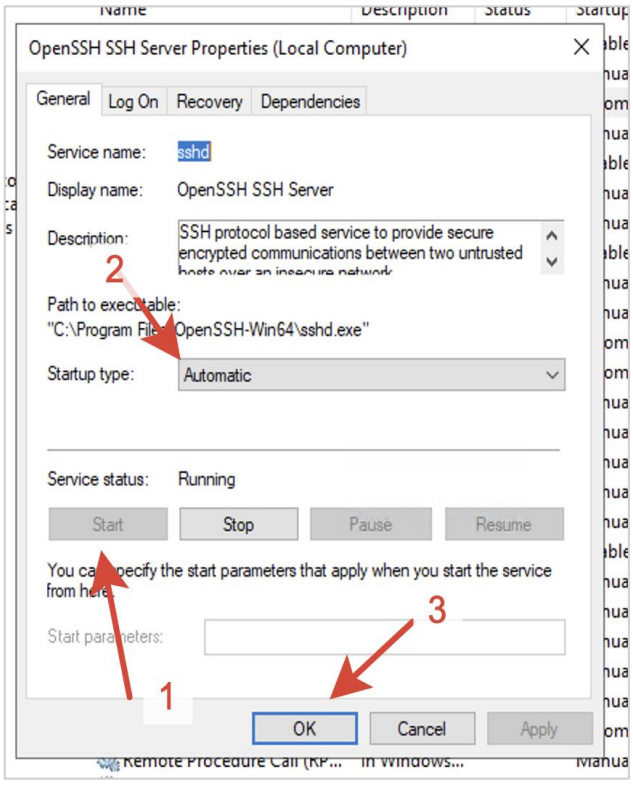

Рисунок 137

- Если сервис не включается, выполняем .\FixHostFilePermissions.ps1 в директории с проектом.
- Делаем правило брандмауэра, пропускающее подключения на 22-й порт (***Рисунок 138*** – ***Рисунок 141***).

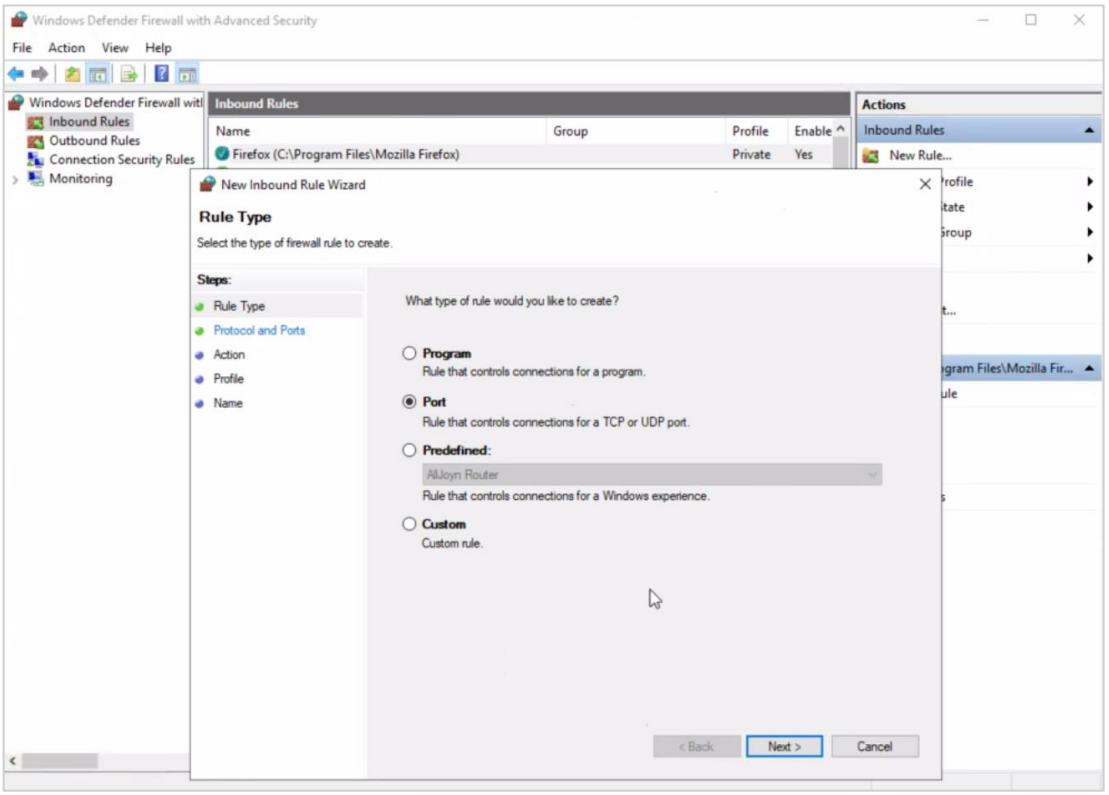

Рисунок 138

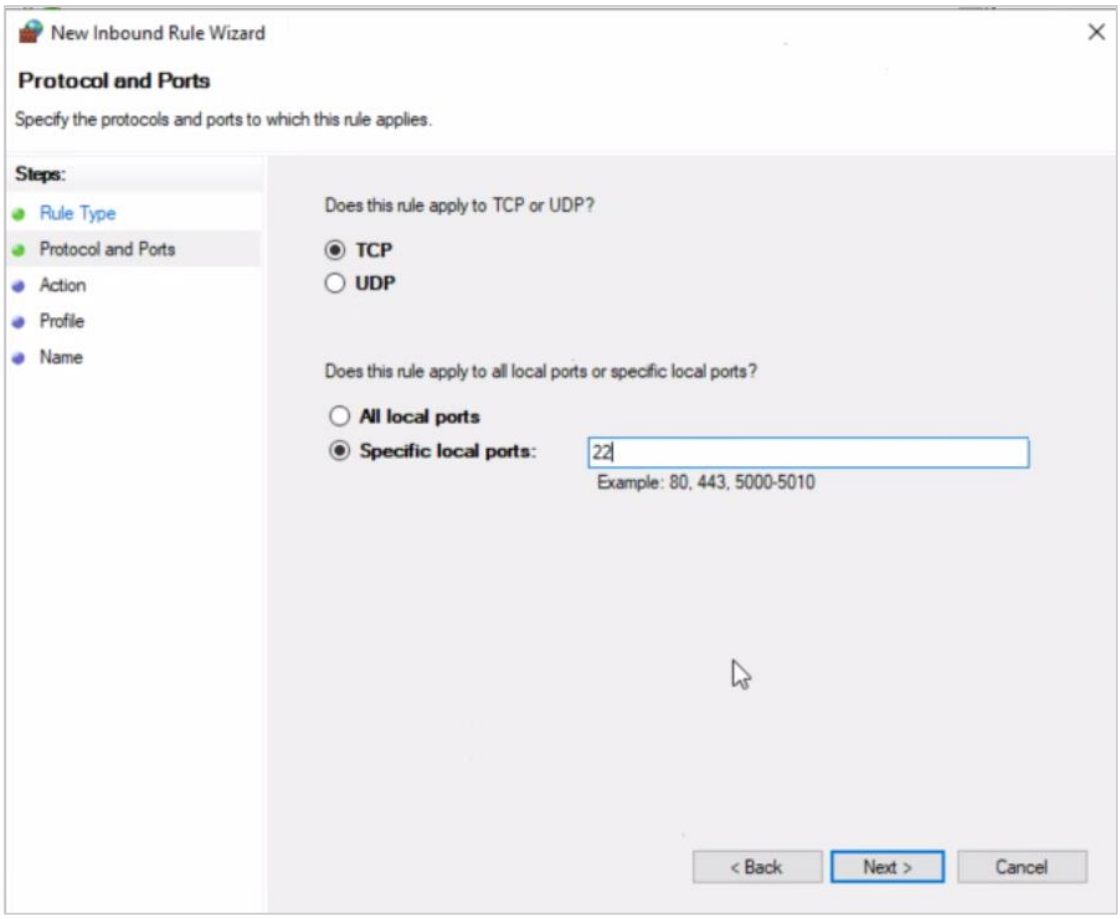

Рисунок 139

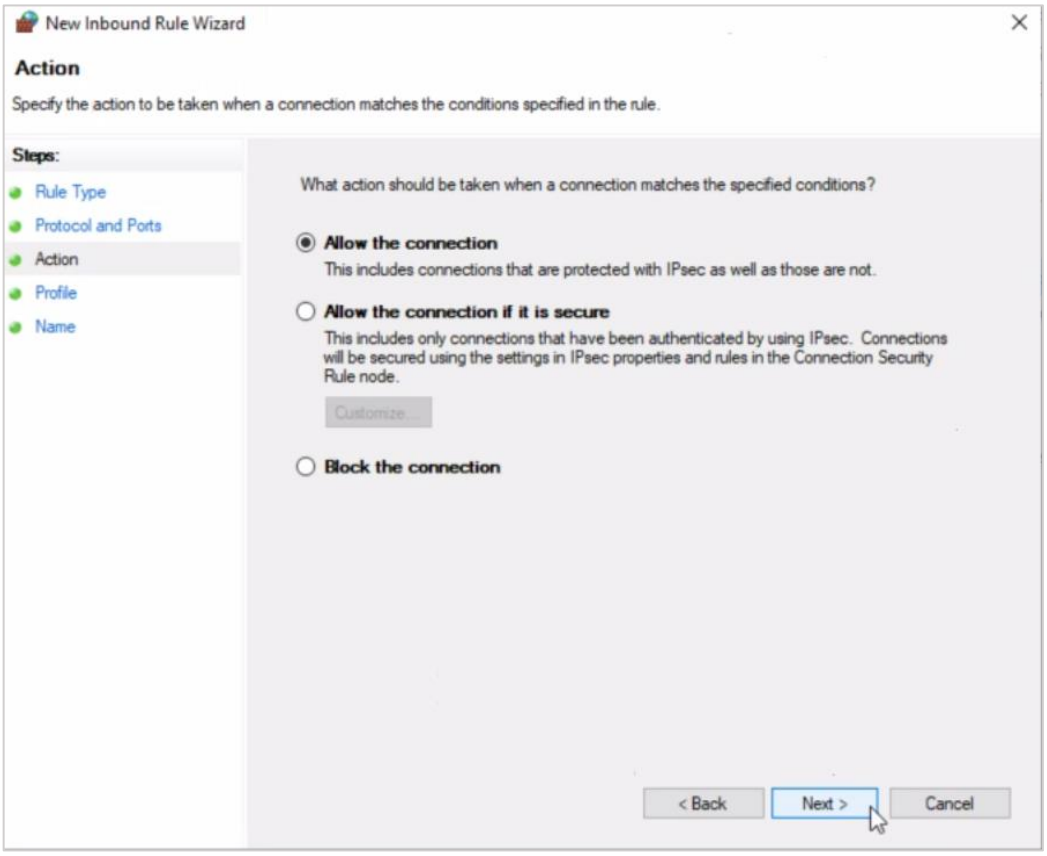

Рисунок 140

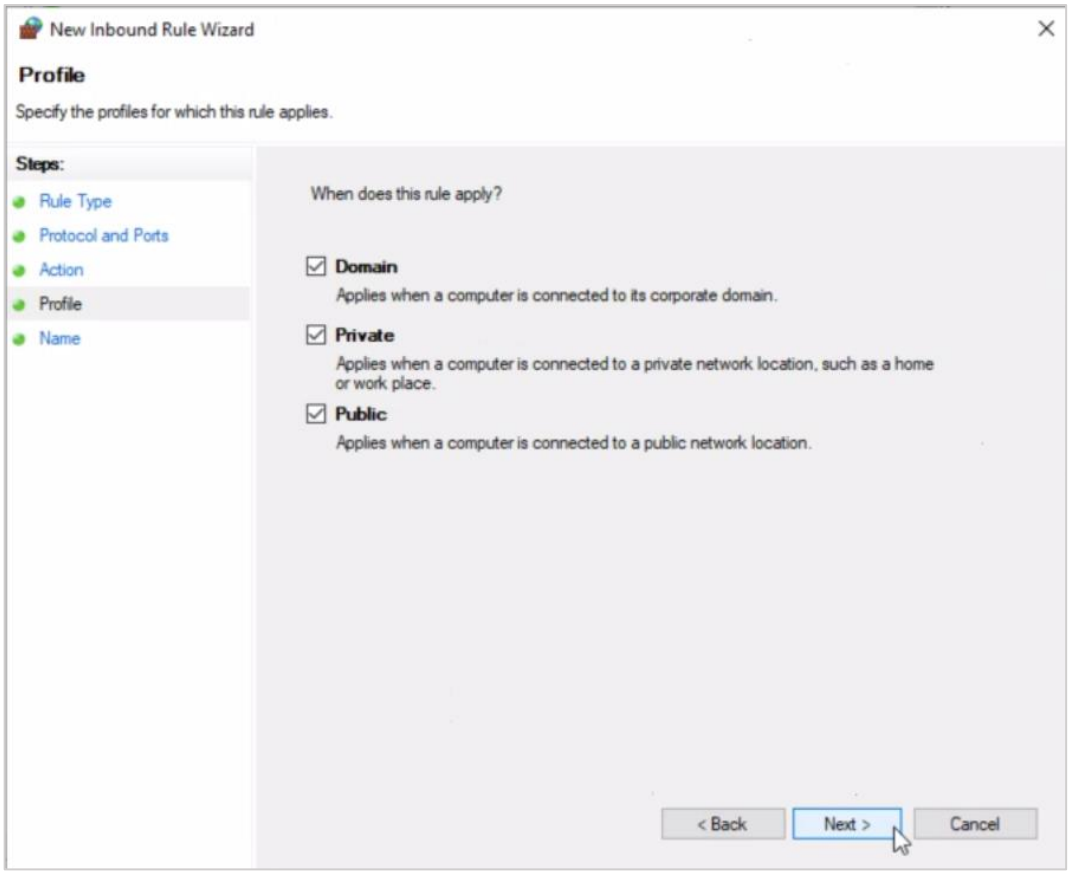

Рисунок 141

- Заходим с ESU-box по SSH на наш сервер и проверяем доступность Powershell-плагина Veeam следующими командами (***Рисунок 142***):

```
powershell
Add-PSSnapin VeeamPSSnapin
Get-PSSnapin VeeamPSSnapin
```

```
~ $ ssh Administrator@10.11.145.251
Administrator@10.11.145.251's password:
Microsoft Windows [Version 10.0.17763.1397]
(c) 2018 Microsoft Corporation. All rights reserved.

administrator@VBR-01 C:\Users\Administrator>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Add-PSSnapin VeeamPSSnapin
PS C:\Users\Administrator> Get-PSSnapin VeeamPSSnapin


Name        : VeeamPSSnapin
PSVersion   : 5.1
Description : This is a PowerShell snap-in that includes the Veeam's cmdlet.
```

Рисунок 142

- Посмотрим, как называются обычные и ScaleOut репозитории, для этого выполним команду:

```
GET-VBRBackupRepository -ScaleOut
```

В выводе команды должен отобразиться настроенный репозиторий (***Рисунок 143***).

```
PS C:\Users\Administrator> Get-VBRBackupRepository -ScaleOut


PolicyType                    : DataLocality
Extent                        : {esu-repo-01, esu-repo-02}
UsePerVMBackupFiles           : True
PerformFullWhenExtentOffline  : False
EnableCapacityTier            : False
OperationalRestorePeriod      : 14
OverridePolicyEnabled         : False
OverrideSpaceThreshold        : 90
OffloadWindowOptions          : 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
                                ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
CapacityExtent                :
EncryptionEnabled             : False
EncryptionKey                 :
CapacityTierCopyPolicyEnabled : False
CapacityTierMovePolicyEnabled : True
Id                            : 13254c30-d943-4148-a543-3add67fdda6d
Name                          : esu-sobr-01
Description                   : esu-rp-01, disk E+F
```

Рисунок 143

Теперь можно сконфигурировать раннер на стороне РУСТЭК-ЕСУ. Необходимо указать хост, логин и пароль администратора сервера, имя ScaleOut backup репозитория и тайм-зону (***Рисунок 144***):

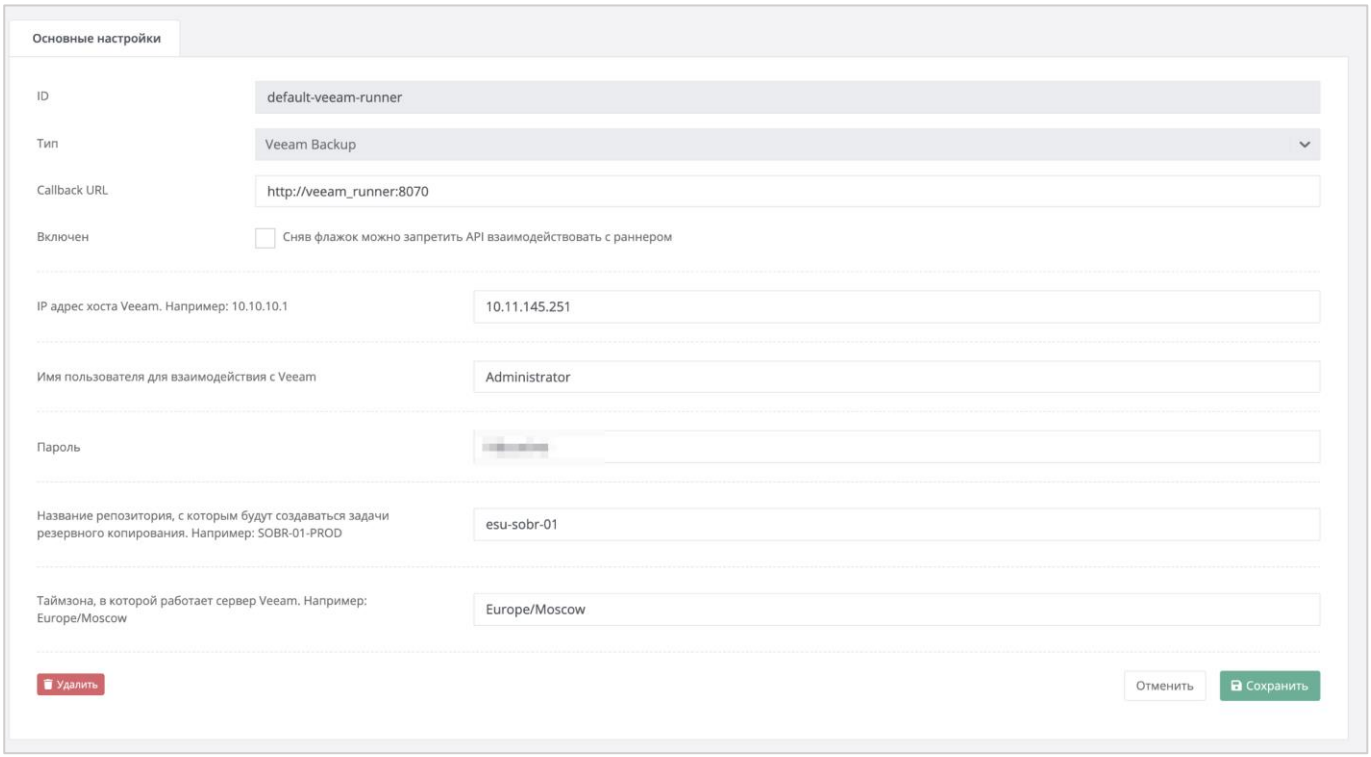

Рисунок 144

12.6. Подключение S3-хранилища на базе NetApp StorageGRID к РУСТЭК-ЕСУ

РУСТЭК-ЕСУ поддерживает интеграцию S3 развёрнутого на базе NetApp StorageGRID.

Примечание: NetApp Storage GRID должен быть развёрнут обязательно, с другими решениями РУСТЭК-ЕСУ интеграцию не поддерживает!

Чтобы подключить хранилище S3 к РУСТЭК-ЕСУ и использовать его из панели управления достаточно произвести настройку S3 раннера.

Для этого необходимо перейти в меню **Инсталляция** – **Система** – **Раннеры**, выбрать S3-runner и в открывшейся форме ввести информацию в соответствующие поля (***Рисунок 145***):

- Адрес API NetApp – указать адрес, по которому доступно API NetApp StorageGRID.
- Имя пользователя-администратора – указать логин администратора NetApp StorageGRID.
- Пароль пользователя-администратора – указать пароль администратора NetApp StorageGRID.
- URL к хранилищу S3 – указать URL по которому доступно S3 хранилище.

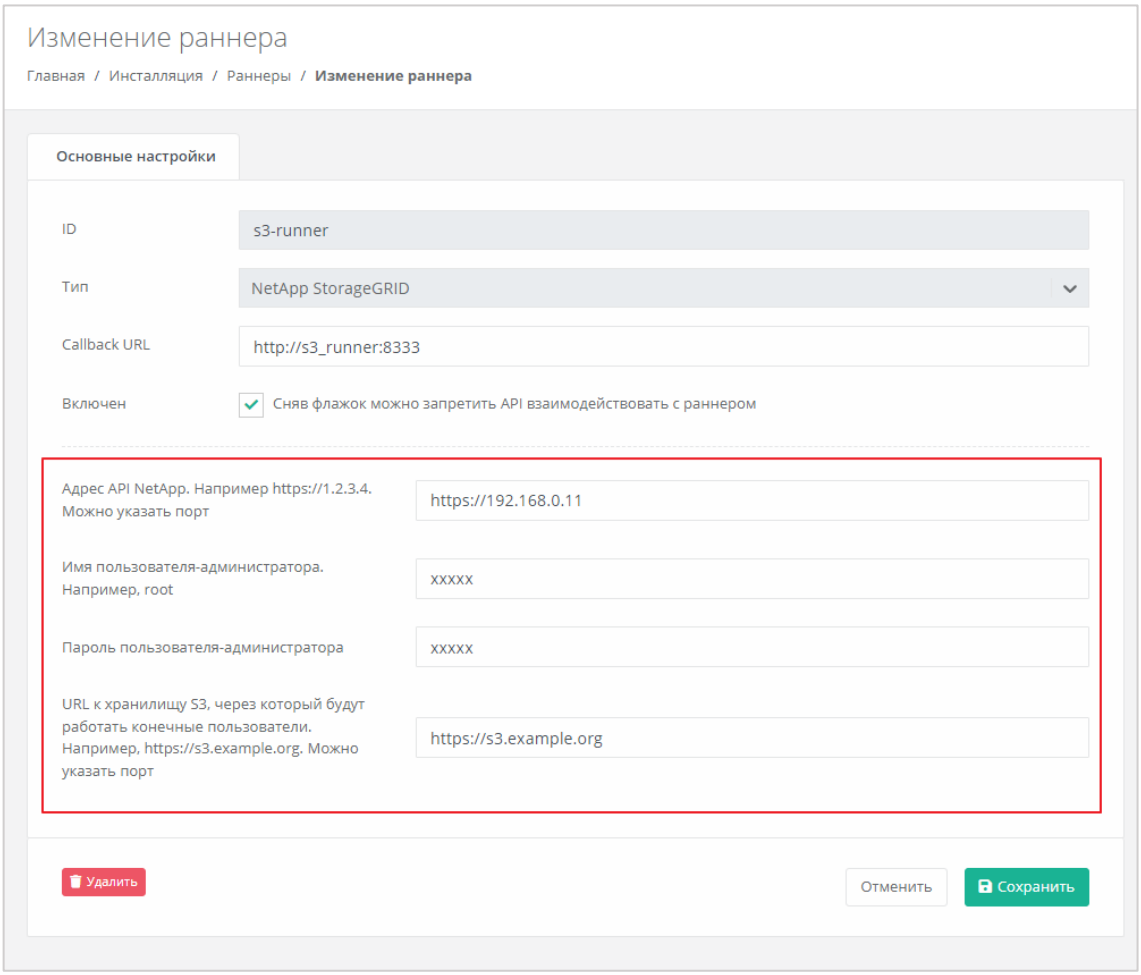

Рисунок 145

После сохранения изменений индикатор S3-раннера должен стать зелёным. После перезагрузки страницы появится пункт **Хранилище S3**.

12.7. Подключение YooKassa к РУСТЭК-ЕСУ

Зачастую, когда инсталляция РУСТЭК-ЕСУ используется в качестве публичного облака, необходимо подключить к ней способы оплаты, с помощью которых клиенты смогут оплачивать заказанные услуги.

Доступные методы оплаты можно задать при создании или изменении клиента в меню **Администрирование** – **Клиенты** (***Рисунок 146***, ***Рисунок 147***).

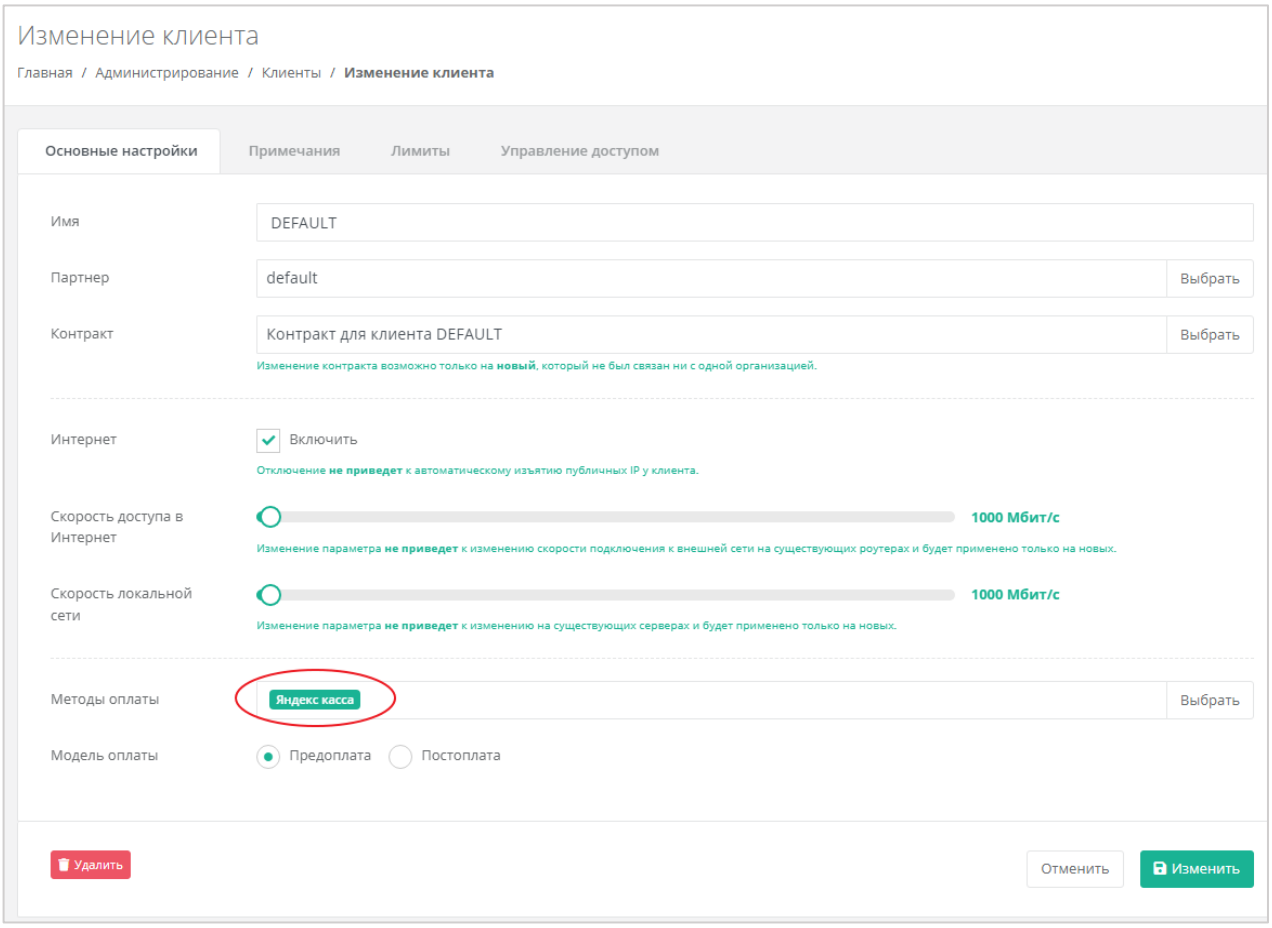

Рисунок 146

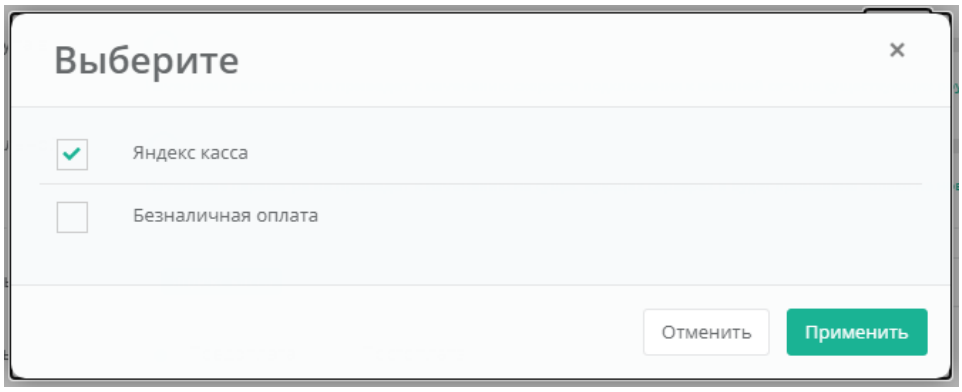

Рисунок 147

РУСТЭК-ЕСУ «из коробки» поддерживает работу с сервисом YooKassa (бывшая ЯндексКасса), но для его работы необходимо произвести некоторые настройки, а именно: указать ID вашего магазина и ваш секретный ключ.

Как их получить описано в официальной документации сервиса: https://yookassa.ru/developers/using-api/interaction-format

После успешного получения ID магазина и секретного ключа можно переходить непосредственно к настройке РУСТЭК-ЕСУ.

Используя ssh, подключаемся по IP адресу к серверу с запущенной РУСТЭК-ЕСУ (ESU-box) и выполняем команду:

```
sudo docker-compose exec api make shell
```

В открывшейся консоли вводим:

```
Setting.objects.create(setting_id='yandex_shop_id', target='paymentmethod-yandex',
value='ваш_id_магазина')

Setting.objects.create(setting_id='yandex_secret_key', target='paymentmethod-yandex',
value='ваш_секретный_ключ')
```

Выходим командой **exit**

Выходим командой **exit**

Далее необходимо произвести настройку HTTP уведомлений в личном кабинете YooKassa. Это необходимо для отправки уведомлений о пополнении в РУСТЭК-ЕСУ (***Рисунок 148***).

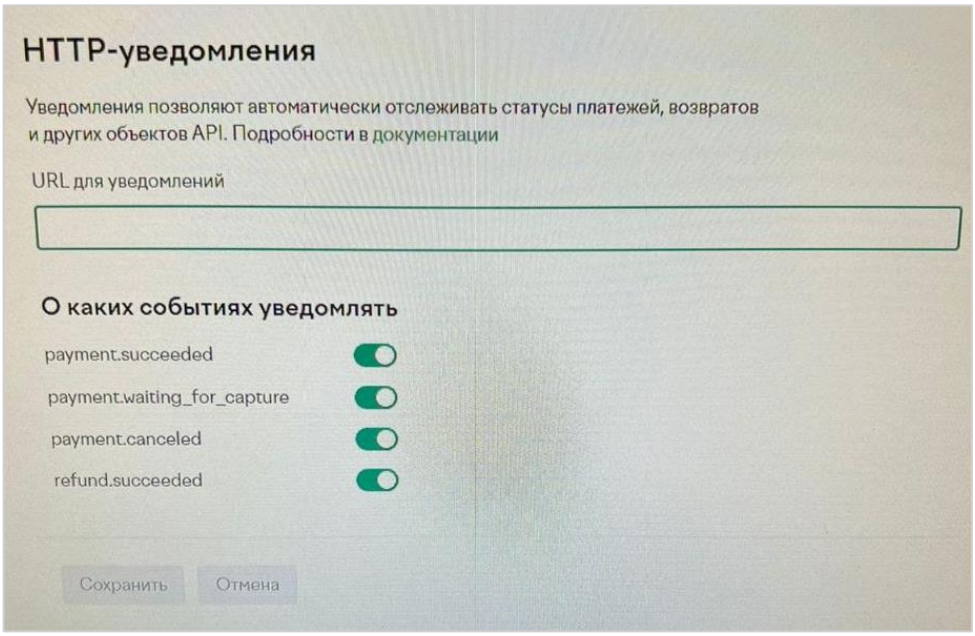

Рисунок 148

URL для уведомлений: https://адрес_API/v1/payment/yandex_payment_callback

На этом настройка завершена.

Примечание: пополнение баланса через сервис YooKassa, возможно только в том случае, если в качестве логина пользователя указан действительный e-mail, в противном случае РУСТЭК-ЕСУ сообщит об ошибке.

Обратите внимание на требования для доставки HTTP уведомлений на официальной странице сервиса: https://yookassa.ru/developers/using-api/webhooks#configuration

Для проверки интеграции настоятельно советуем сначала подключить тестовый магазин:

https://yookassa.ru/developers/payment-acceptance/testing-and-going-live/testing

13. Развёртывание на платформе виртуализации VMware vSphere

В инструкции описан процесс установки и настройки РУСТЭК-ЕСУ на платформе виртуализации РУСТЭК / KVM, данный способ является предпочтительным и рекомендуемым, но продуктом также поддерживается установка на платформу виртуализации VMware vSphere.

13.1. Системные требования

Для развёртывания на платформе виртуализации VMware vSphere необходимы:

- VMware vSphere (6.7, 7.0);
- dvSwitch и сервисная портгруппа, одна маршрутизируемая подсеть не меньше /27 с доступом до сетей хостов VMware и Vcenter

Необходимые работы на стороне VMware для подключения к РУСТЭК-ЕСУ:

1. Создать пользователя esu-admin с правами администратора.
2. Создать Datacenter.
3. Создать кластером хоста(ов) в Datacenter, внутри которого будут создаваться ВМ и edge-роутеры.
4. Создать Datastore Cluster из датастора(ов), на котором будут размещаться пользовательские edge-роутеры и служебные сервисы.
5. Создать Datastore Cluster из датастора(ов), на котором будут размещаться диски пользователей (можно использовать из пункта **4**).
6. Создать dvSwitch, под которым будут создаваться пользовательские сети (порт-группы).

13.2. Порядок развертывания

Создаём management-сеть РУСТЭК-ЕСУ – портгруппу на dvSwitch в vSphere (требуется один VLAN). Необходимо учитывать, что в эту сеть будут подключены пользовательские роутеры для сегмента VMware и что сеть должна быть маршрутизируемой.

Таким образом, размер подсети напрямую влияет на максимальное число ВЦОДов. Сервер с установленной РУСТЭК-ЕСУ (ESU-box) станет DHCP-сервером в этой подсети.

Заводим маршрутизируемую сеть внутрь dvSwitch в vSphere, в данном примере она называется ESU_management_vlan3235_n10.11.14.0m24, VLAN ID 3235 (***Рисунок 149*** – ***Рисунок 152***).

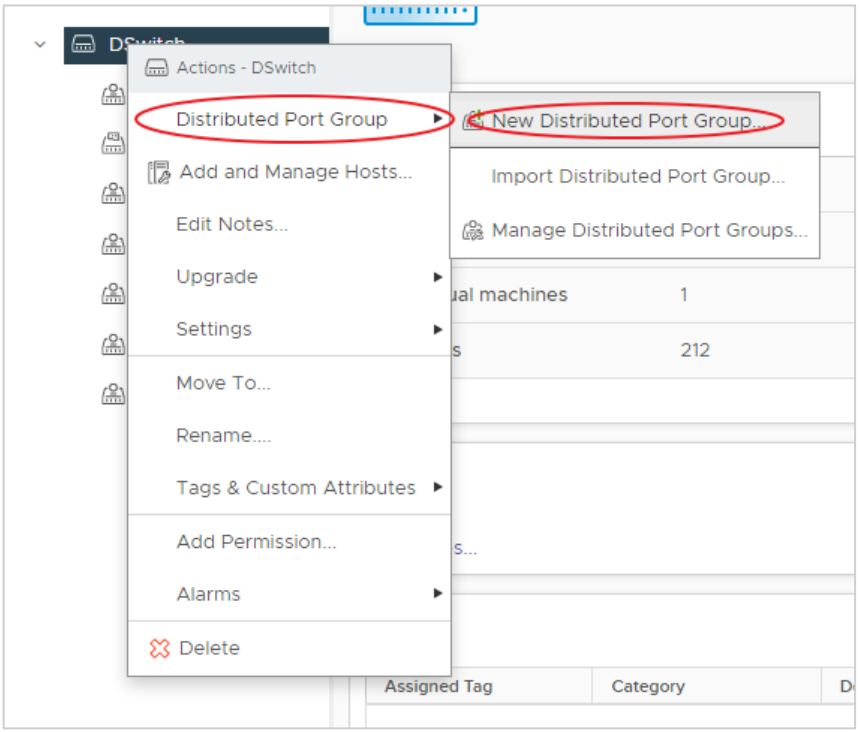

Рисунок 149

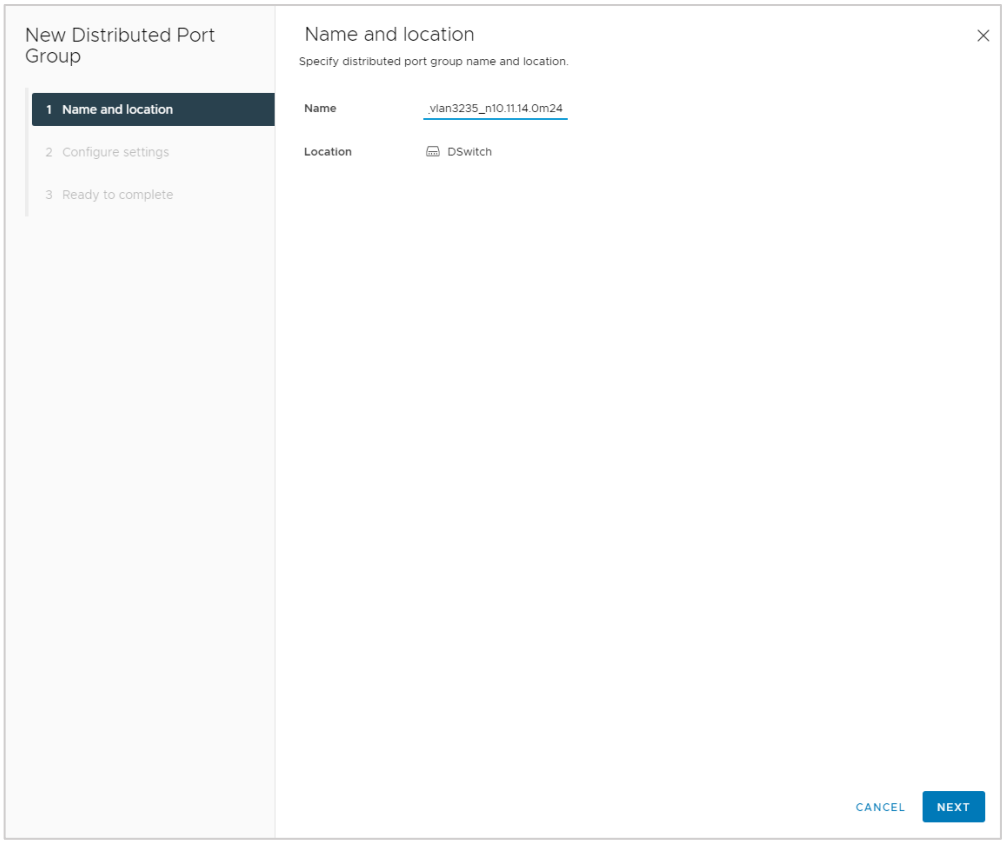

Рисунок 150

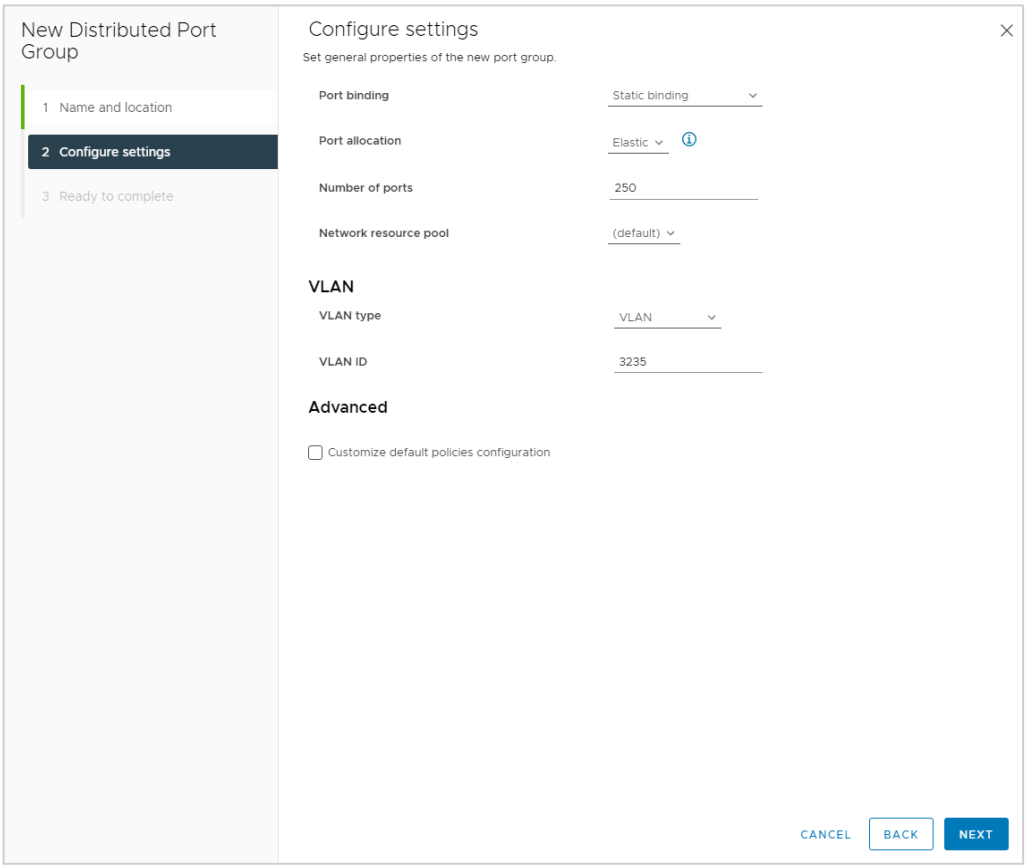

Рисунок 151

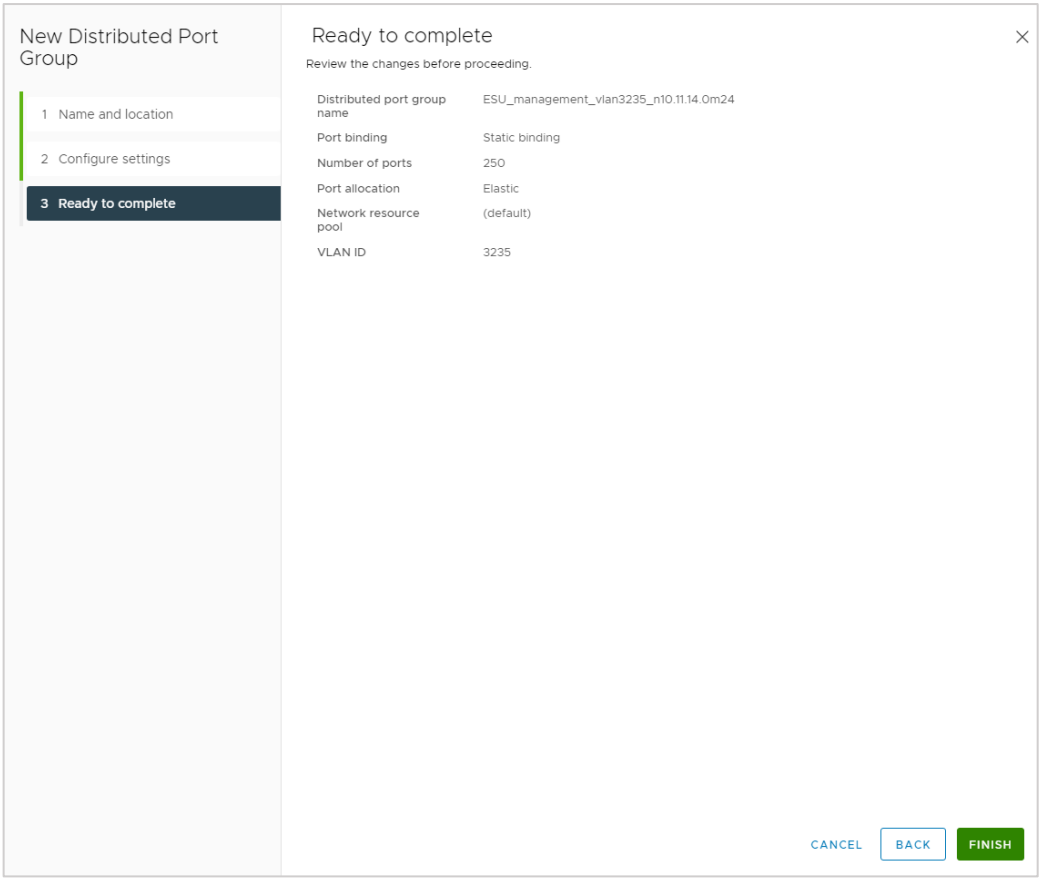

Рисунок 152

Переходим в редактирование созданной портгруппы и удостоверимся, что параметры указаны в соответствии с указанными ниже (***Рисунок 153*** – ***Рисунок 154***).

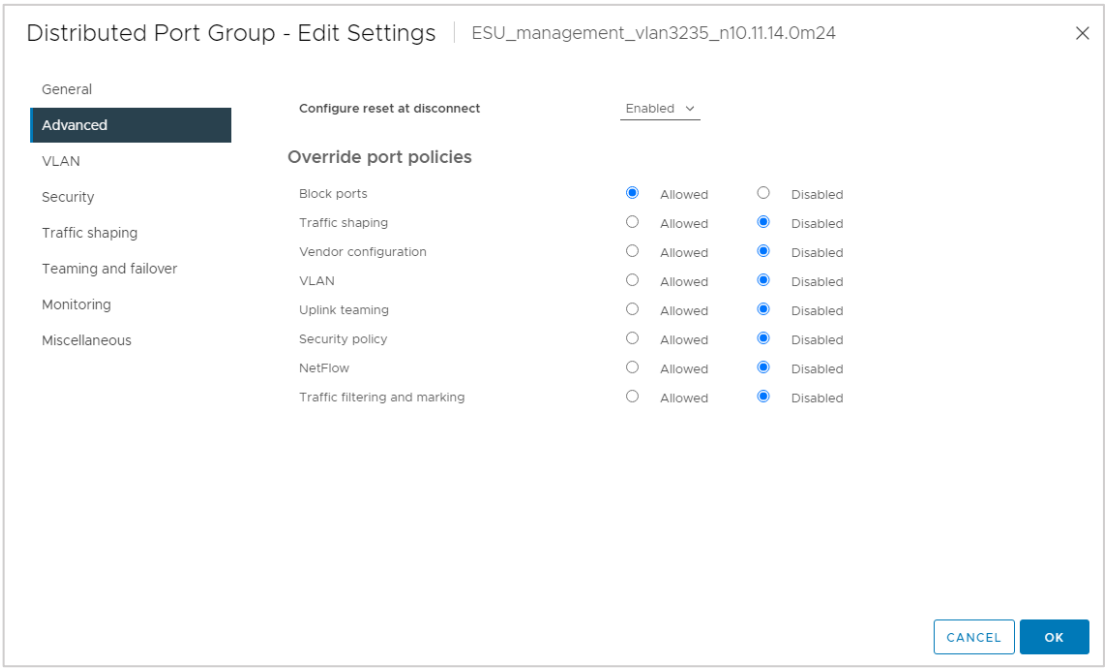

Рисунок 153

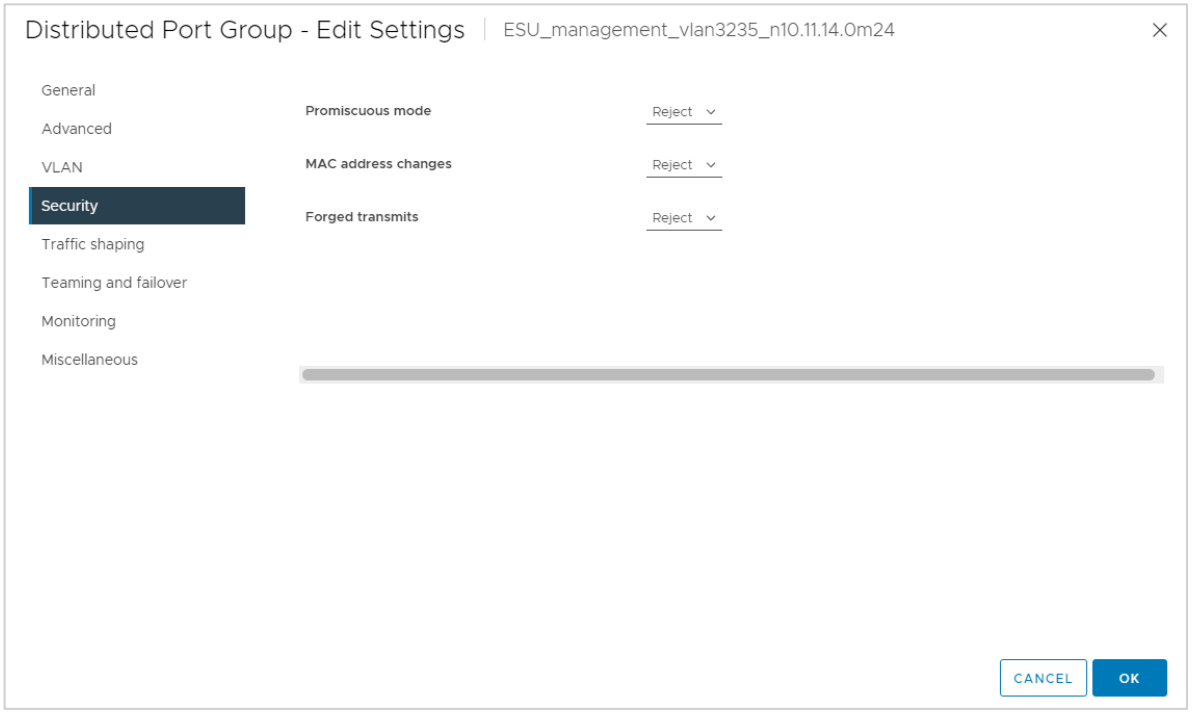

Рисунок 154

Создаём директорию, в которой будут расположены ВЦОДы клиентов и сама РУСТЭК-ЕСУ (ESU-box). Например, ESU3, а в ней создадим папку Management (***Рисунок 155*** – ***Рисунок 158***):

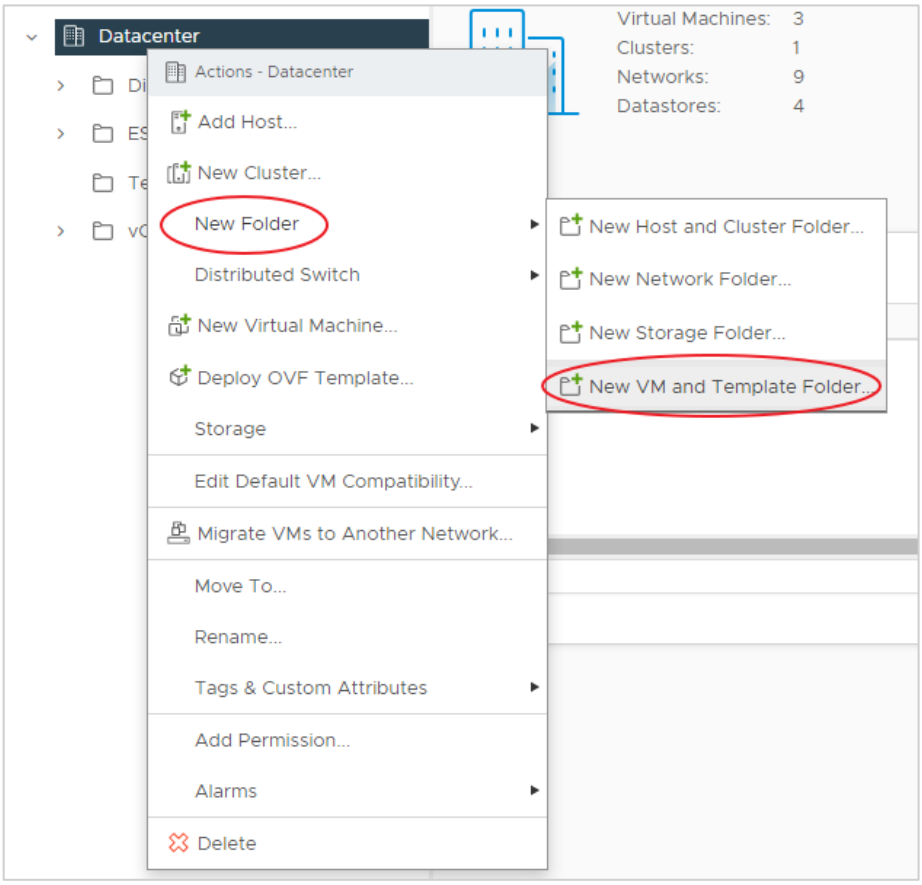

Рисунок 155

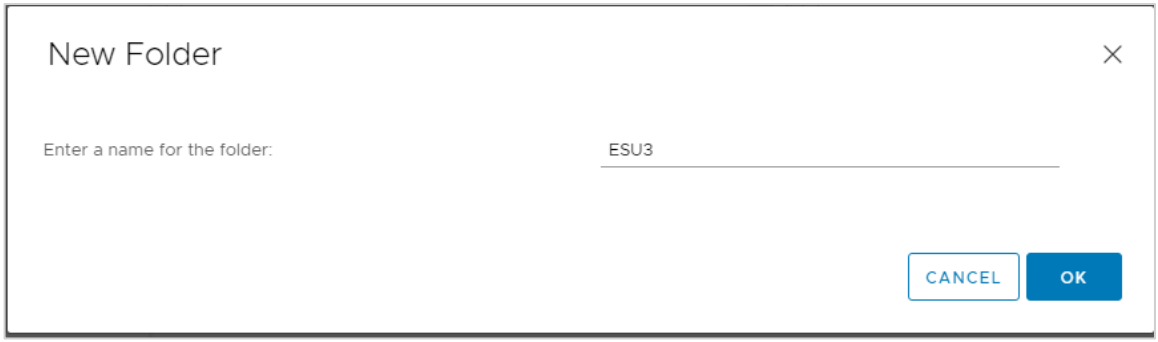

Рисунок 156

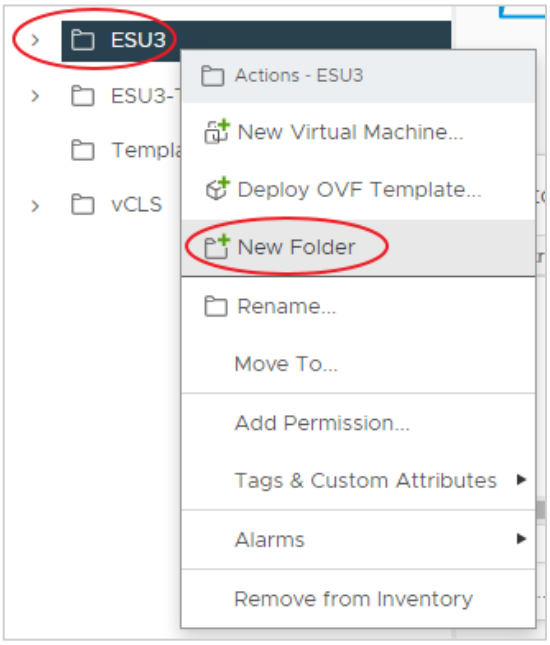

Рисунок 157

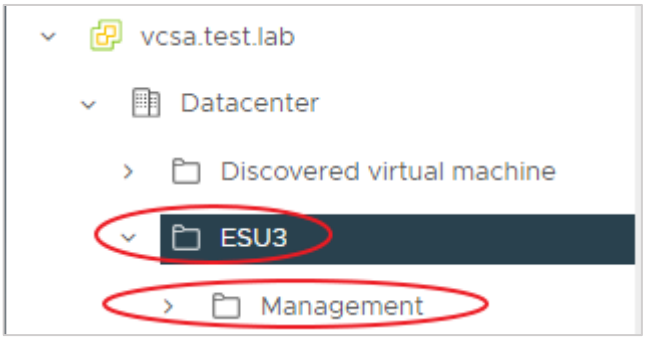

Рисунок 158

Далее необходимо загрузить предоставленный образ сервера с РУСТЭК-ЕСУ (ESU-box) в vSphere. Для этого выбираем папку Management и нажимаем «Deploy OVF Template» (***Рисунок 159***).

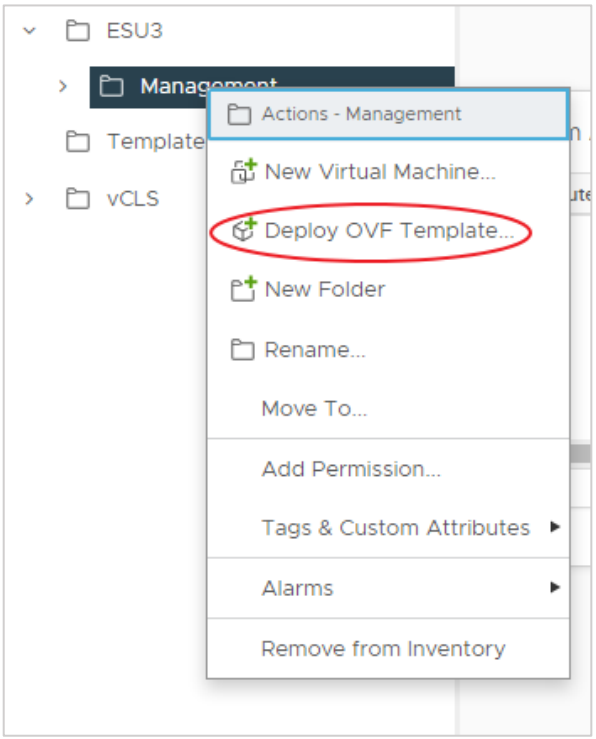

Рисунок 159

Далее выбираем предоставленный .ova-образ (***Рисунок 160***).

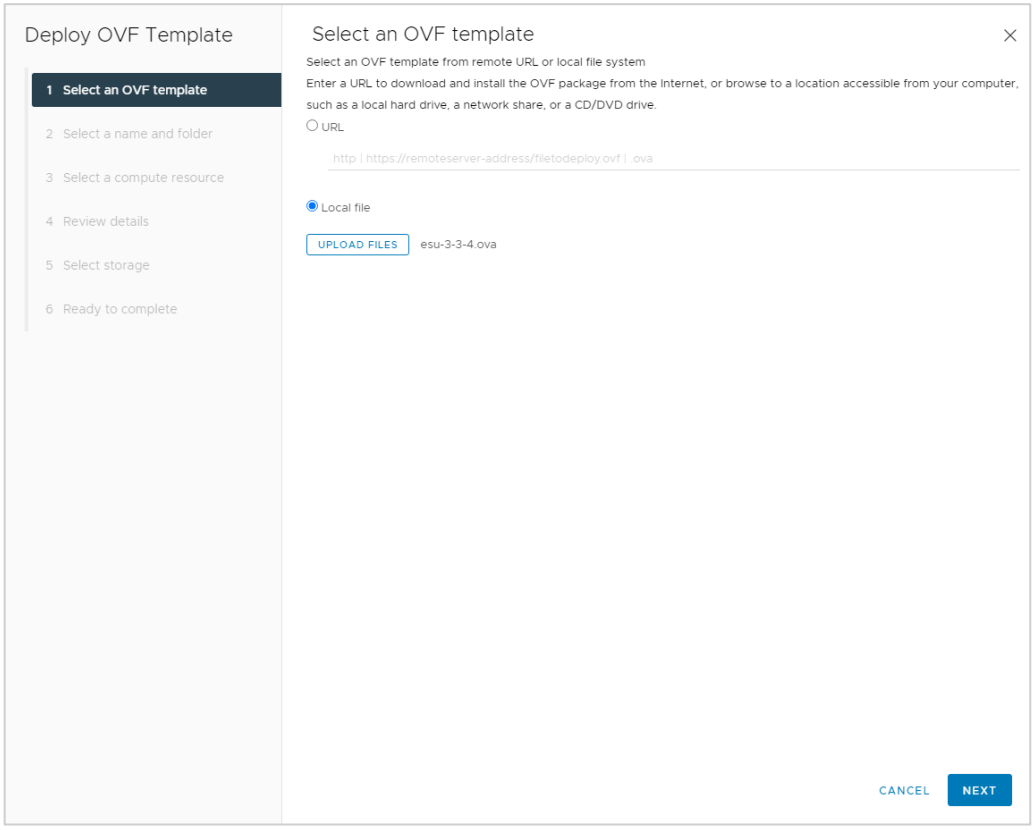

Рисунок 160

Выбираем созданную папку Management, где будет развёрнут сервер (***Рисунок 161***).

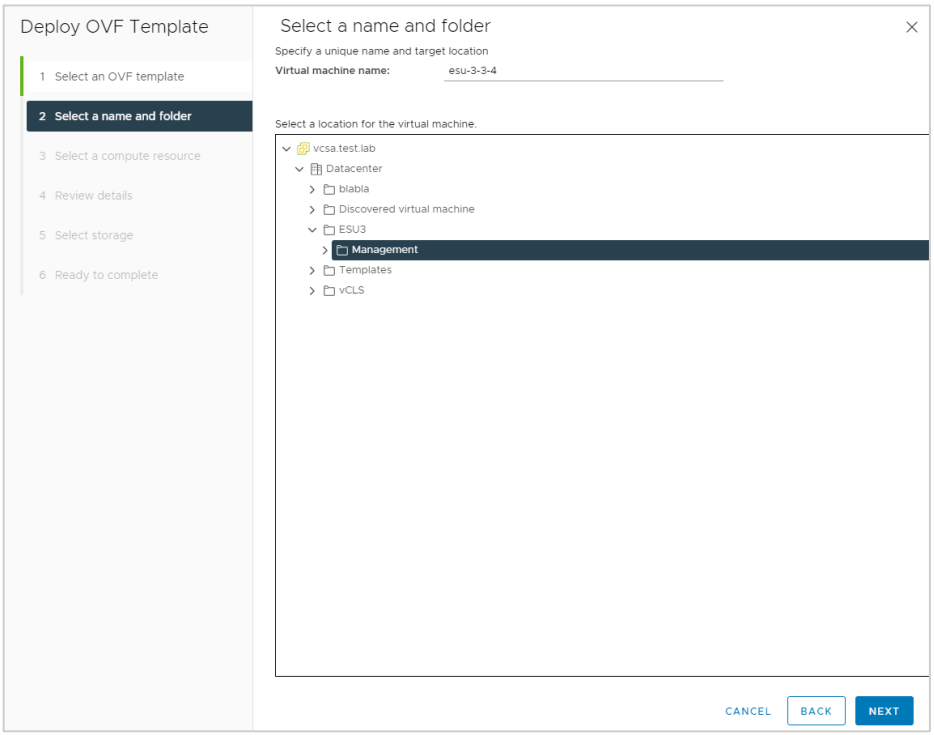

Рисунок 161

Выбираем кластер, где будет развёрнут сервер (***Рисунок 162***).

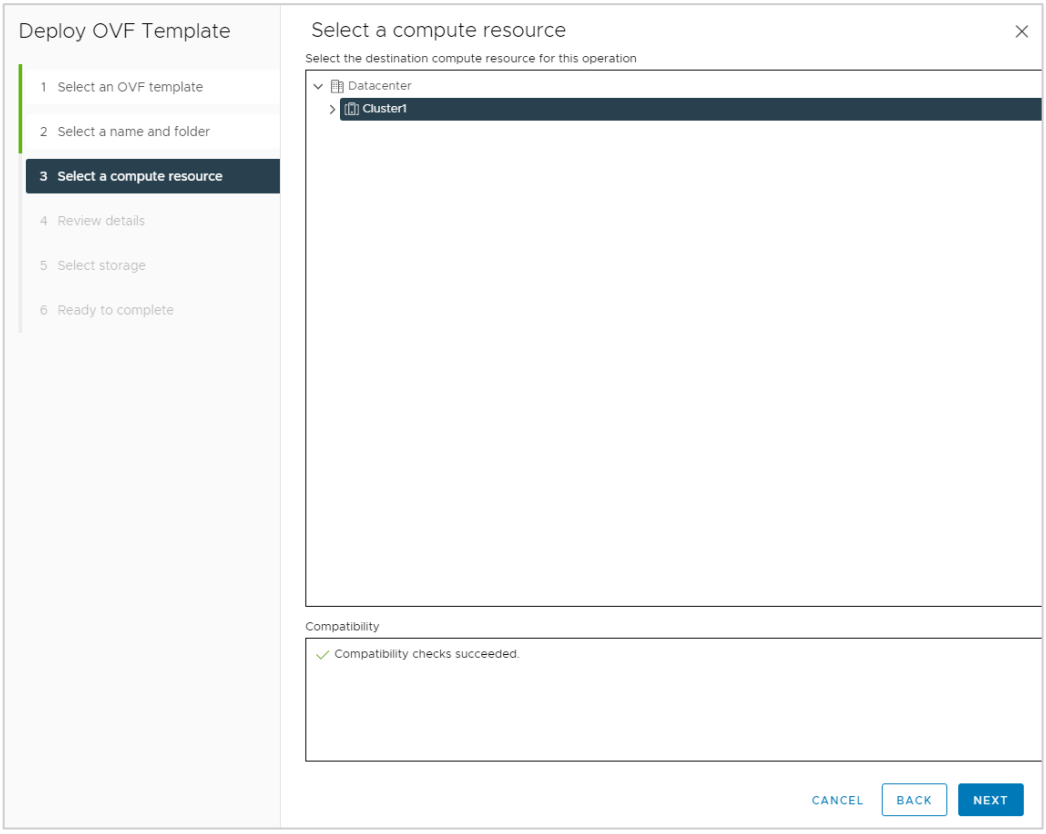

Рисунок 162

Подтверждаем дальнейшие действия (***Рисунок 163***).

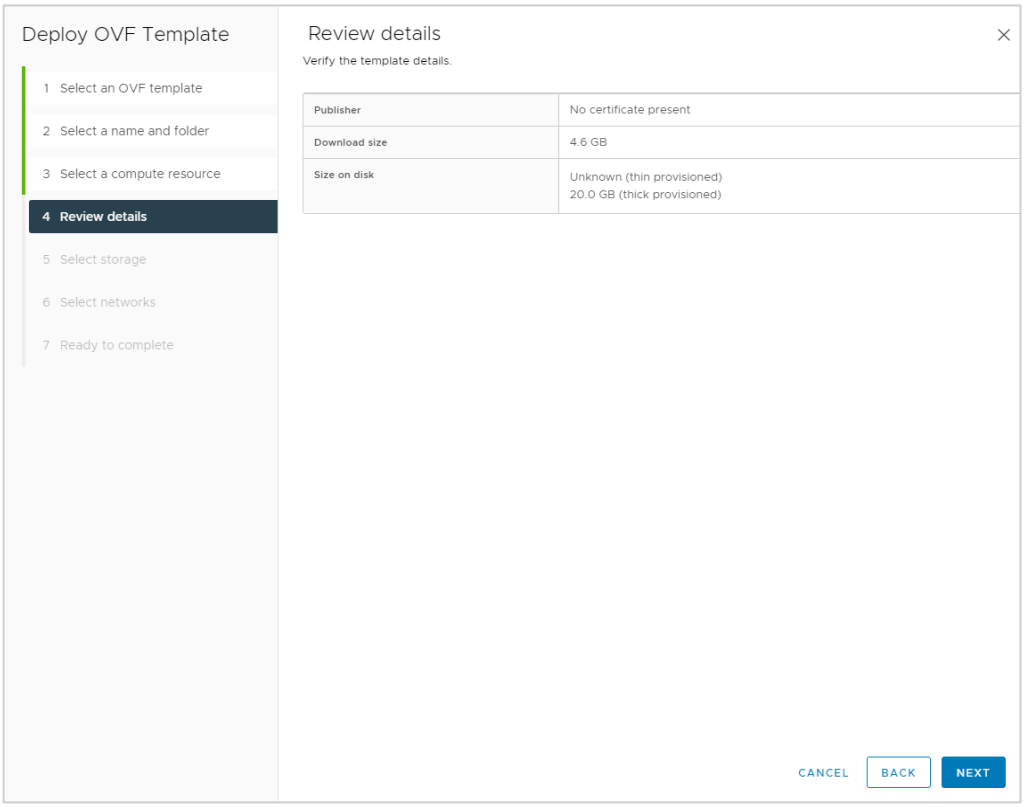

Рисунок 163

Выбираем формат диска Thin Provision и датастор для диска сервера (***Рисунок 164***).

Thin Provision должен быть выбран обязательно!

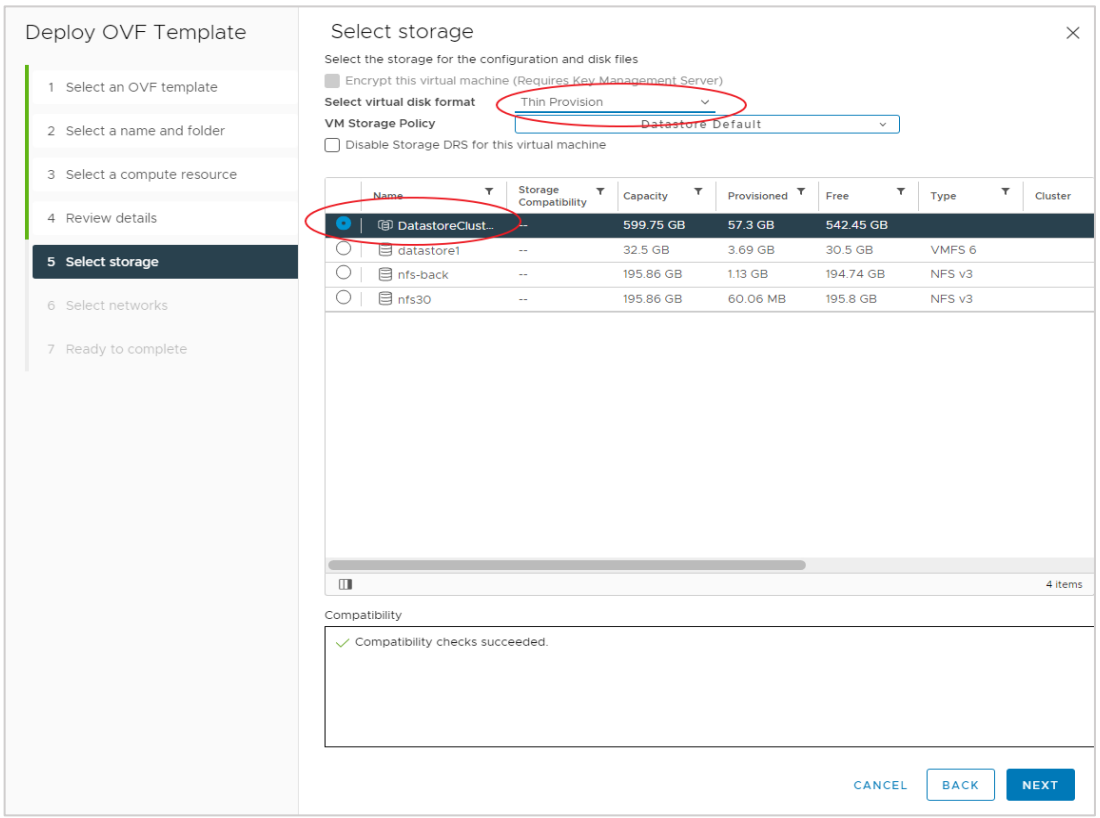

Рисунок 164

Выбираем сеть, которая будет подключена к нашему серверу. Выбираем созданную раннее и заведённую в dvSwitch портгруппу, в следующем окне жмём «FINISH» (***Рисунок 165***, ***Рисунок 166***).

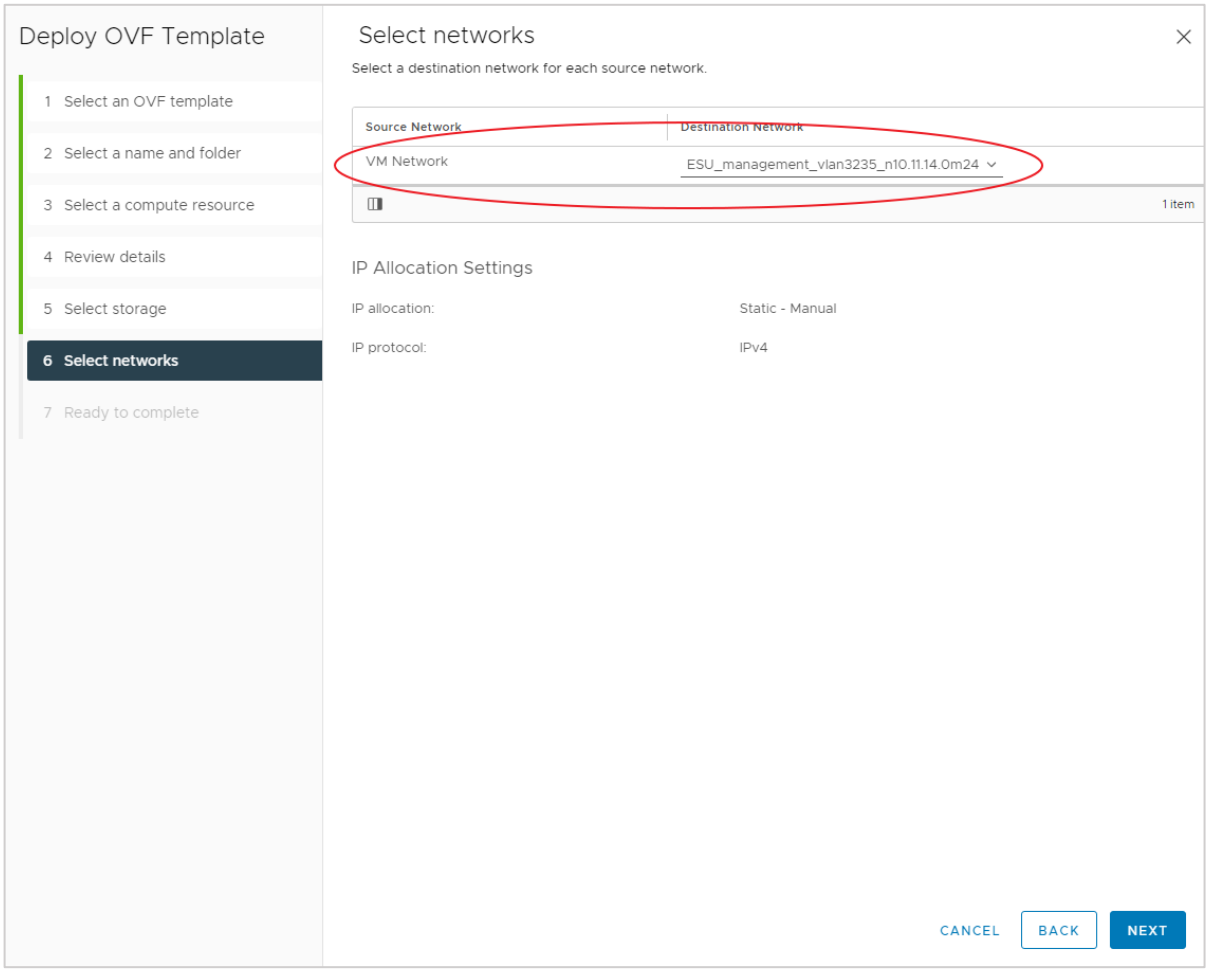

Рисунок 165

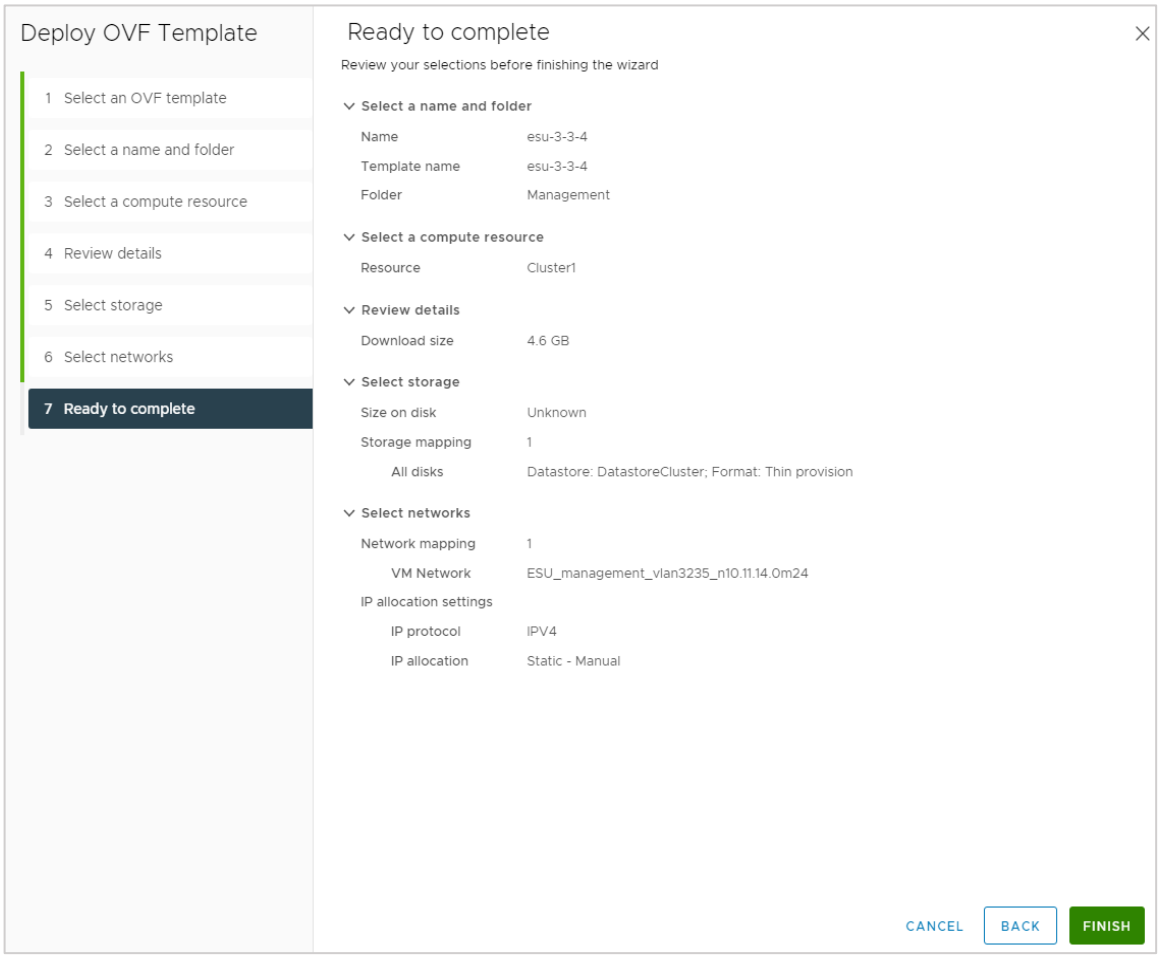

Рисунок 166

Начнётся процесс развёртывания (***Рисунок 167***).

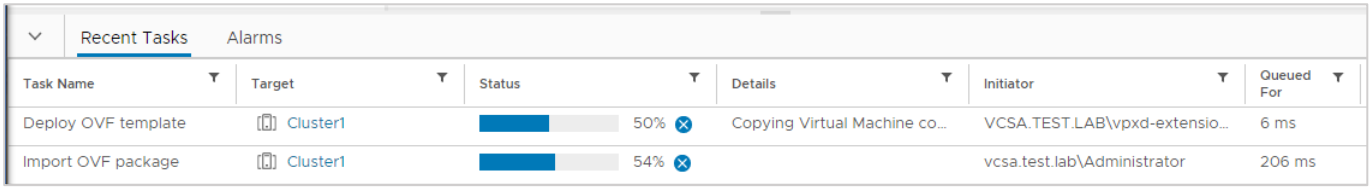

Рисунок 167

После развёртывания включаем сервер и открываем консоль (***Рисунок 168***).

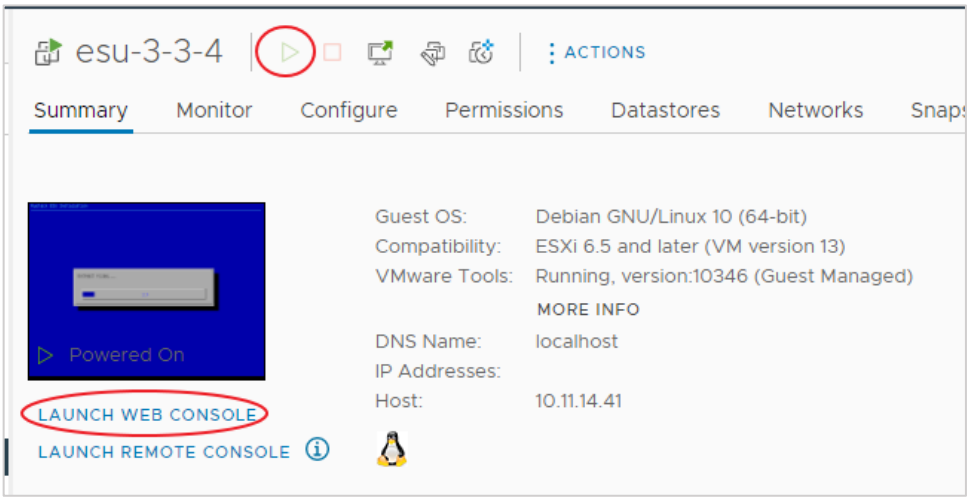

Рисунок 168

Стандартная учётная запись на сервере с РУСТЭК-ЕСУ (ESU-box): **deploy:1-qpALzm/**

13.3. Примечания по установке и дальнейшей настройке

- Процесс установки аналогичен установке на платформе виртуализации РУСТЭК (см. раздел **2**). Но на этапе выбора IP адреса необходимо выбрать адрес внутри заведённой в dvSwitch портгруппы**.** Адрес должен быть выделен заранее (см. ***Рисунок 22***).
- Панель управления РУСТЭК-ЕСУ будет доступна по адресу, указанному при установке.
- Сервер с РУСТЭК-ЕСУ (ESU-box) будет доступен по SSH по адресу, указанному при установке.
- До настройки ресурсного пула РУСТЭК/KVM в панели управления РУСТЭК-ЕСУ необходимо завести внешнюю сеть и подсеть для неё в платформу виртуализации РУСТЭК. Процесс создания внешней сети и подсети описан в пунктах (см. раздел **2.2,** пункты **6**, **7**).
- Для создания кластеров Kubernetes в сегменте РУСТЭК/KVM (см. раздел **11**) в панели РУСТЭК необходимо завести аналогичную портгруппе в dvSwitch, сеть. Далее процесс настройки одинаков для обоих случаев. Процесс создания сети и подсети в РУСТЭК описан в пунктах (см. раздел **2.2,** пункты **4**, **5**). Безопасность портов и DHCP должны быть отключены.

Остальные настройки производятся аналогично ситуации, когда РУСТЭК-ЕСУ развёрнута на платформе виртуализации РУСТЭК.

14. Подготовка инфраструктуры для получения обновлений РУСТЭК-ЕСУ

Обновления РУСТЭК-ЕСУ выпускаются примерно раз в месяц.

Для того, чтобы получать эти обновления и использовать всегда актуальную версию, необходимо настроить свою инфраструктуру таким образом, чтобы эти обновления было возможно доставить.

Обновление производится службой поставщика продукта с помощью Gitlab-раннера, установленного на стороне заказчика по согласованию с ним.

- Gitlab-раннер может быть установлен на сервер с РУСТЭК-ЕСУ (ESU-box) или на отдельный сервер на базе OS Linux, с которого по SSH доступен ESU-box.
- Сервер с установленным Gitlab-раннером должен иметь исходящий доступ во внешнюю сеть Интернет по протоколу HTTPS – это необходимо для установки связи между Gitlab на стороне поставщика продукта и Gitlab-раннером для доставки обновлений.

Сценарий подготовки к получению обновлений:

- Установить Gitlab-раннер согласно официальной документации: https://docs.gitlab.com/runner/install/.
- Предоставить доступ по SSH к серверу ESU-box и к серверу с установленным Gitlab-раннером для проведения процедуры регистрации Gitlab-раннер и авторизации ESU-box в Docker Registry.
- Инженерная служба поставщика продукта осуществляет процедуру регистрации Gitlab-раннера с помощью сгенерированного токена. Процедура описана в официальной документации: https://docs.gitlab.com/runner/register/.
- Инженерная служба поставщика продукта осуществляет процедуру авторизации ESU-box в Docker Registry с помощью сгенерированной пары login/password.
- Выполнить команду:

```
sudo docker login -u [user] -p [password] docker.vds2b.com
```

- После проведения процедуры регистрации раннера и авторизации сервера ESU-box (сценарий авторизации ESU-box описан ниже), SSH-доступ можно отключить.
- Дальнейшие настройки для получения обновлений производятся службой поставщика продукта на стороне Gitlab.

Сценарий авторизации ESU-box:

- Установка Gitlab-раннера на стороне заказчика в ESU-box / сервер рядом.
- Генерация токена для регистрации раннера и пары логин/пароль для авторизации ESU-box в Docker Registry на нашей стороне.
- Выдача этих пар заказчику.
- Регистрация раннера с выданной парой логин/пароль на стороне заказчика.
- Авторизация ESU-box в Docker Registry с выданной парой логин/пароль на стороне заказчика.

Приложение 1 Пример Auto DevOps-скрипта

```
from vdc.models import FirewallTemplate, FirewallRule
from rest_framework import serializers



def check(vm):
    if not vm.floating:
        raise serializers.ValidationError('Для правильного запуска необходимо назначить публичный IP для этого сервера')



def on_start(vm):
    # Force to enable "Allow Web" rule
    allow_web_rule = FirewallTemplate.objects.get_or_none(name='Разрешить WEB', vdc=None)
    if allow_web_rule and vm.floating:
        for port in vm.ports.filter(type='vm_int'):
            port.fw_templates.add(allow_web_rule)
```