Настройка SMP

- Первый вход и 2FA
- Добавление лицензии с функционалом XDR
- Создание пользователя в SMP
- Замена сертификата консоли OSMP

Первый вход и 2FA

Первый вход и двухфакторная аутентификация (2FA)

После успешной установки откройте веб-браузер и введите URL консоли `https://console.<smp_domain>`. Например, `https://console.snd.example.local/`.

> Необходимо использовать именно доменное имя, а не IP-адрес

1. Войдите под пользователем **admin** с заданным **password**.

2. Должно появиться новое окно 2FA с предложением использовать любое приложение-аутентификатор на мобильном устройстве. Для этого можно использовать секретный ключ или QR-код. После успешного импорта учётной записи в приложение 2FA введите защитный код в поле ниже.

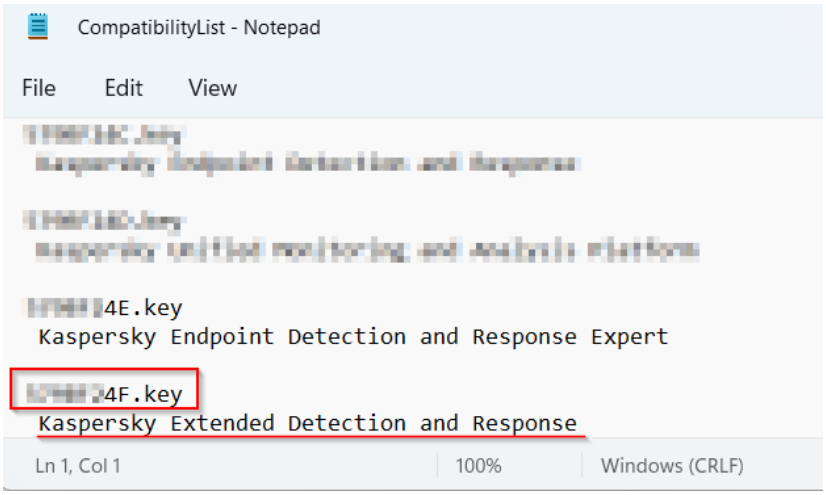
3. Если всё в порядке, откроется окно с руководством по усилению защиты. Внимательно прочитайте его и подтвердите согласие. Затем нажмите **Accept**.

Добавление лицензии с функционалом XDR

> Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

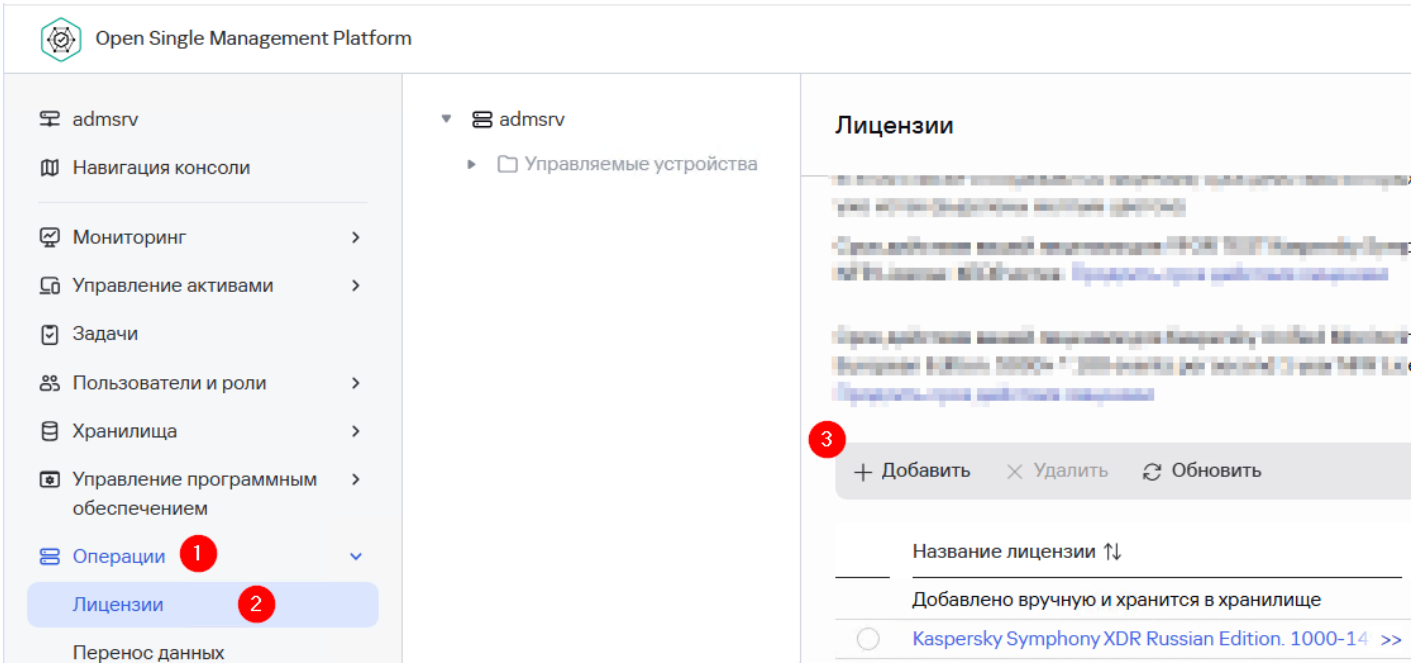
После установки SMP для доступа к функционалу XDR (алерты, инциденты с графом расследования, плейбуки и прочее) необходимо добавить лицензию на функционал. Для этого необходимо выполнить следующее:

1. В полученном архиве с лицензионными ключами найти файл `CompatibilityList.txt` и в нем найти название ключа, которое соответствует функционалу **Kaspersky Extended Detection and Response** (не путать с Kaspersky Endpoint Detection and Response!)

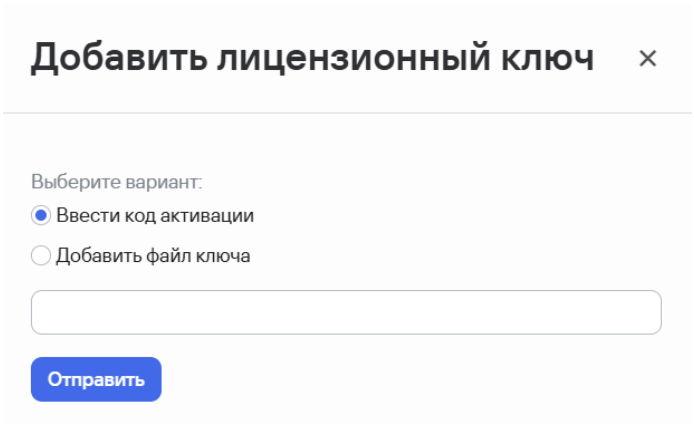


2. Зайти в веб-интерфейс консоли под встроенной учетной записью `admin`, либо под другой с аналогичными правами.

3. Перейти в раздел **Операции** - **Лицензии "Лаборатории Касперского"** и нажать "**Добавить**"

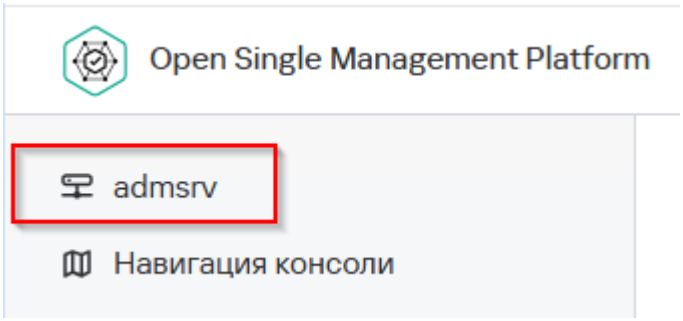


4. В появившемся окне можно либо ввести код активации, который доступен в архиве с лицензиями в файле `ActivationCodes*.txt`, либо загрузить файл ключа, полученный на шаге 1.

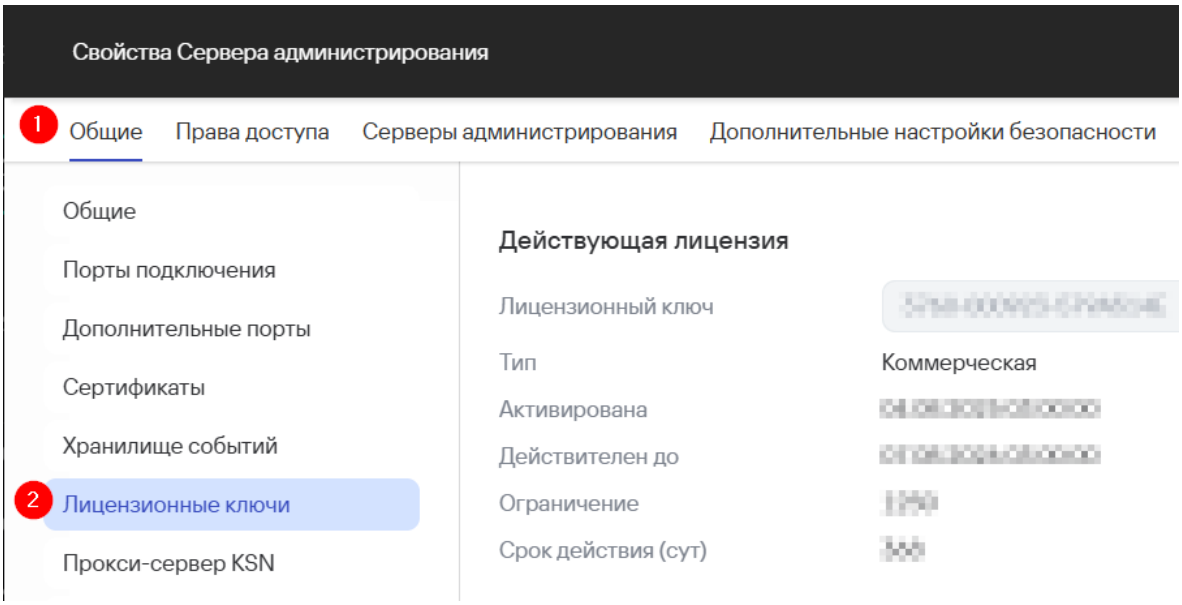


После добавления ключ отобразится в таблице с лицензиями.

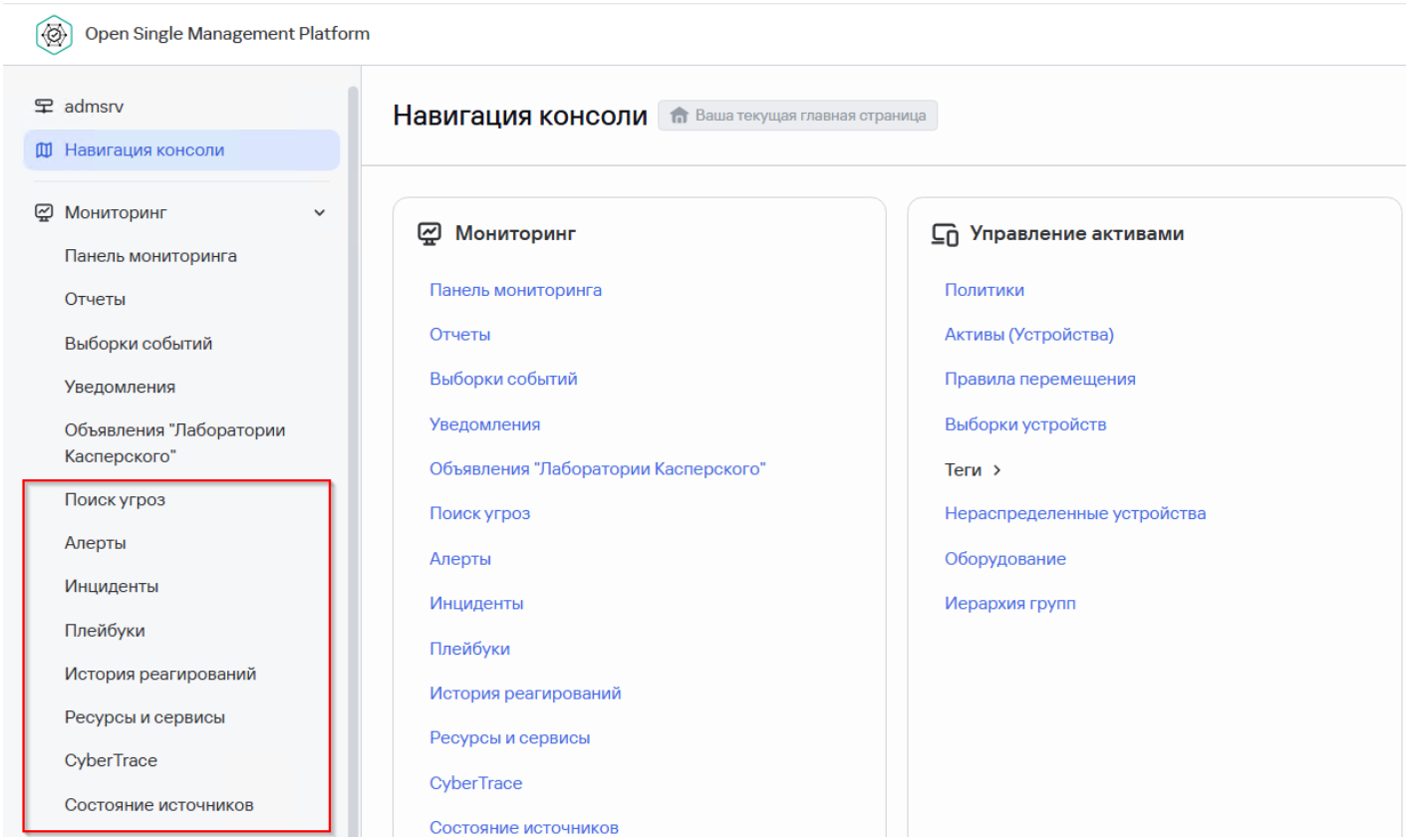
5. Далее необходимо перейти в настройки щелкнув по соответствующему значку в верхней части интерфейса



6. Затем перейти в раздел **Общие** - **Лицензионные ключи** и выбрать ранее добавленный ключ



7. После этого необходимо сохранить все изменения, подождать несколько минут и перезагрузить страницу. Новые разделы будут добавлены в консоль автоматически в раздел "**Мониторинг и отчеты**".



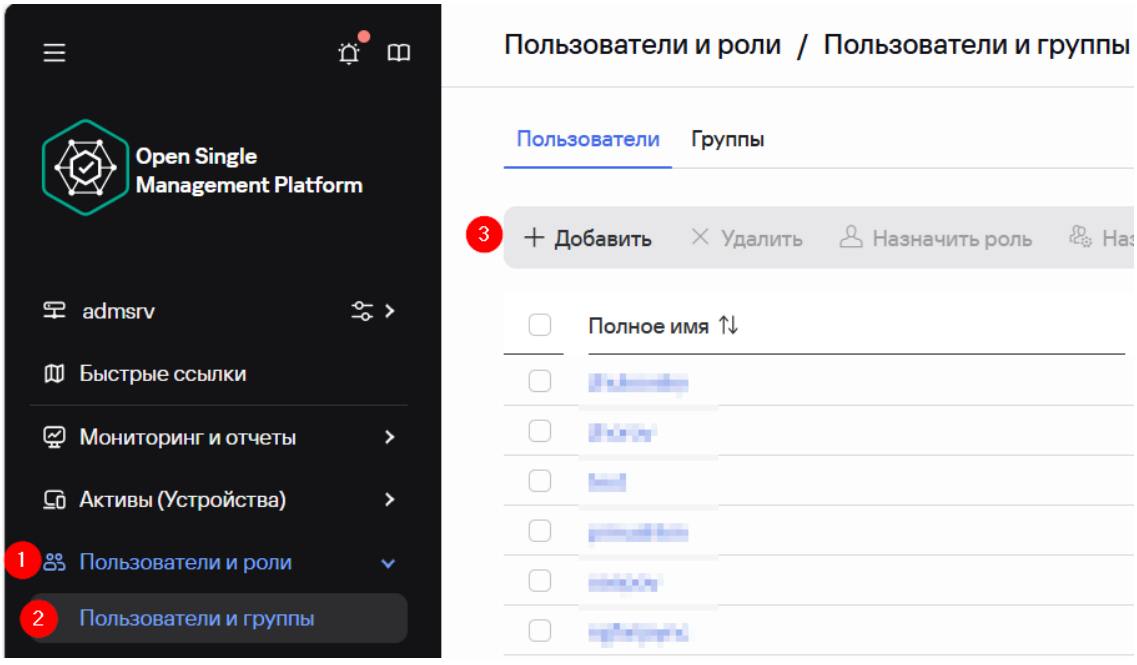
Создание пользователя в SMP

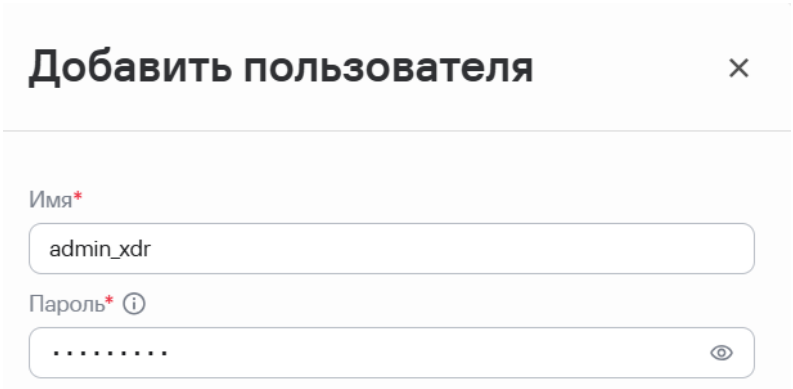
> Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Процесс создания пользователя в SMP аналогичен процессу создания пользователя в KSC. Однако, для предоставлению пользователю доступа к функционалу XDR и консоли KUMA потребуются дополнительные действия.

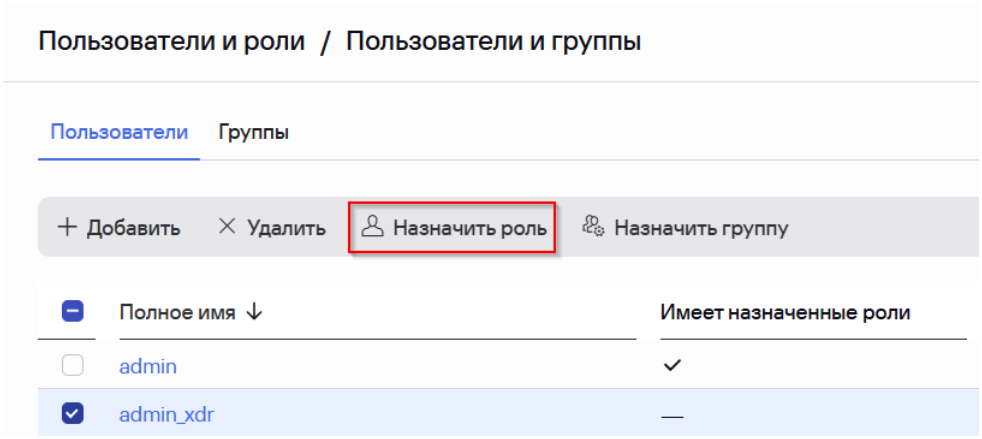
Для создания пользователя и назначения ему прав для доступа ко всему функционалу XDR необходимо выполнить следующие действия

1. Перейдите в интерфейс SMP на вкладку **Пользователи и роли** - **Пользователи и группы** и нажмите **Добавить** для добавления локального пользователя и укажите его логин и пароль (пропустите этот шаг для доменного пользователя).

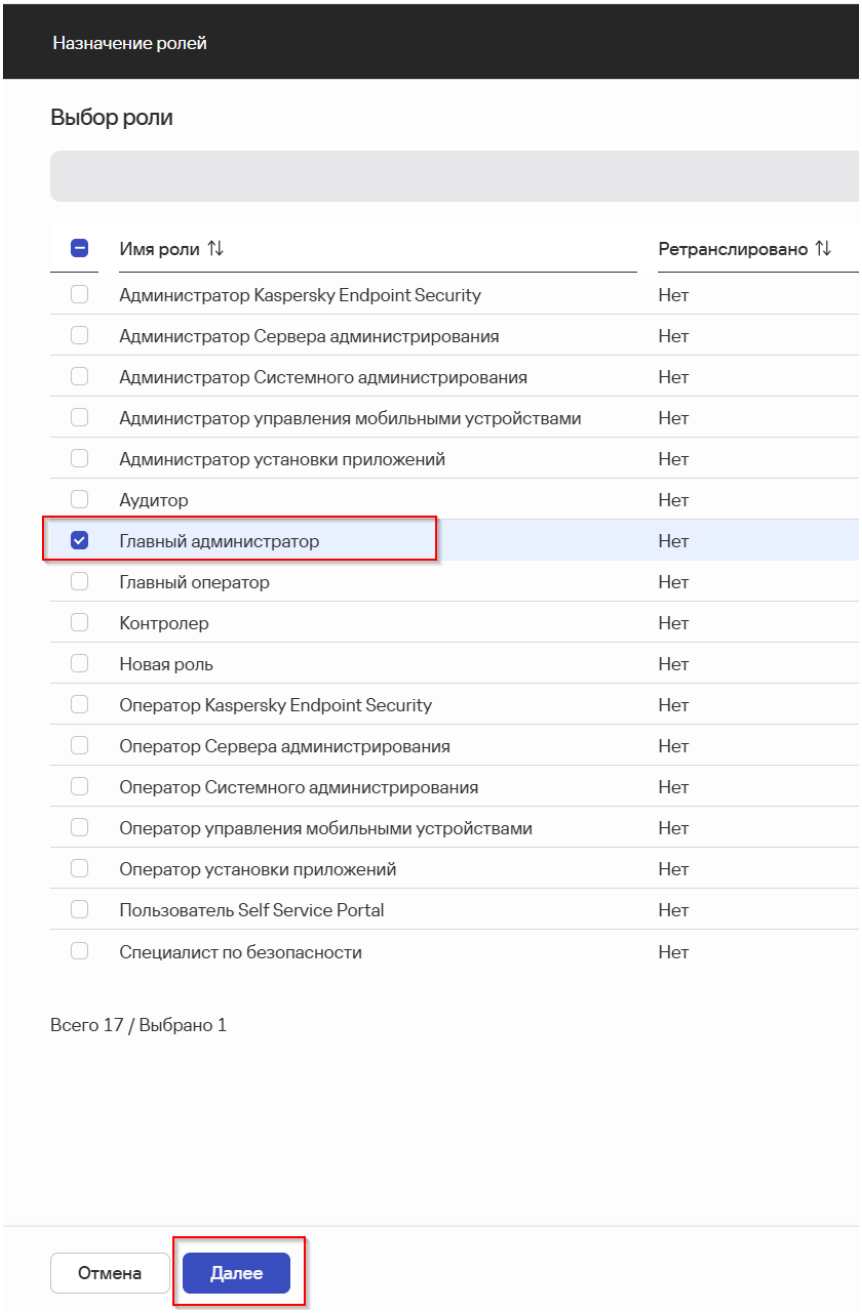




2. После добавления локального пользователя (либо при наличии доменного, полученного через интеграцию с AD) необходимо назначить роль. Для этого выберите галочкой нужного пользователя и нажмите **Назначить роль**.



3. Выберите необходимую роль галочкой и нажмите далее

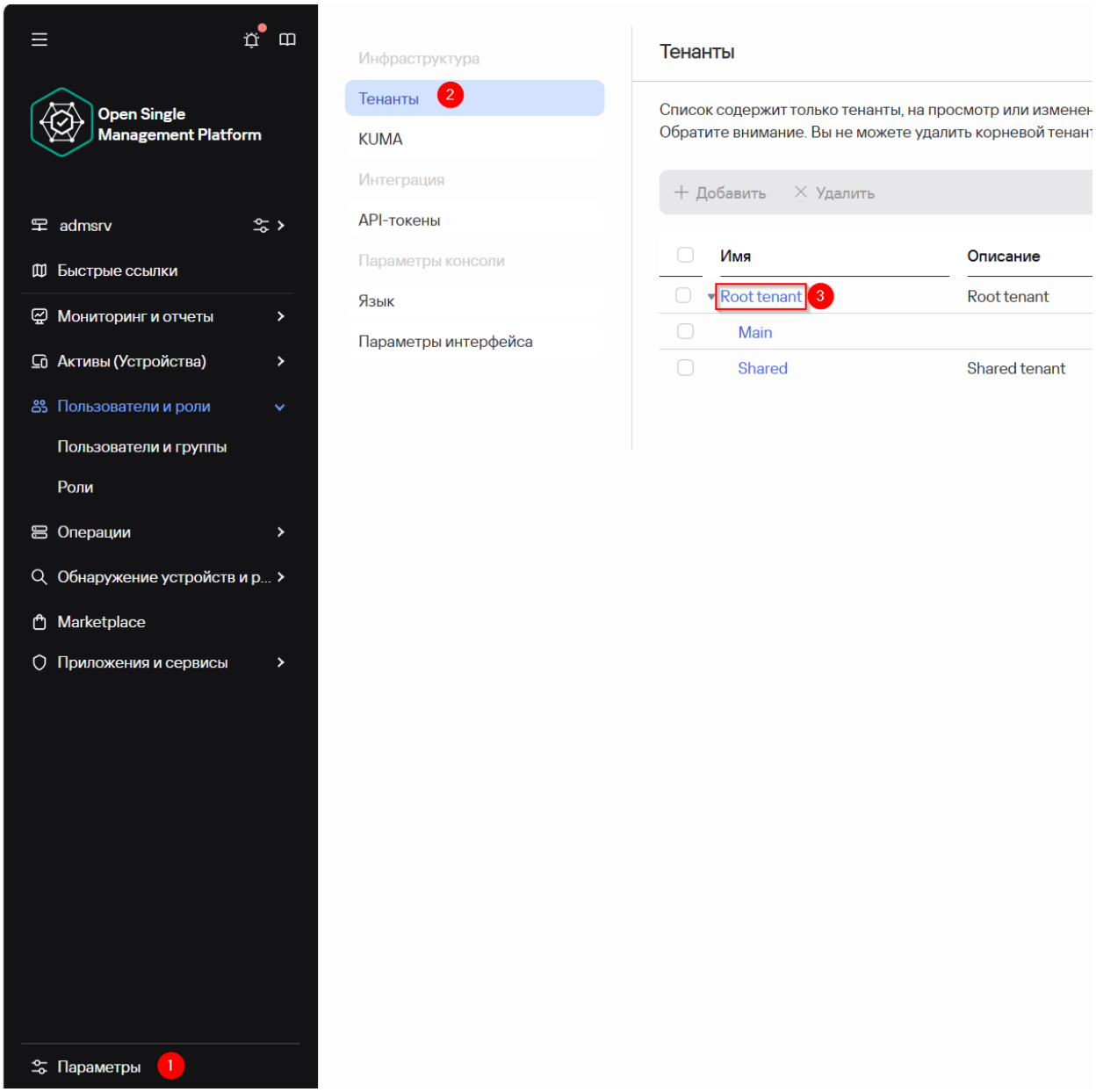


4. Выберите область действия и нажмите **Готово**

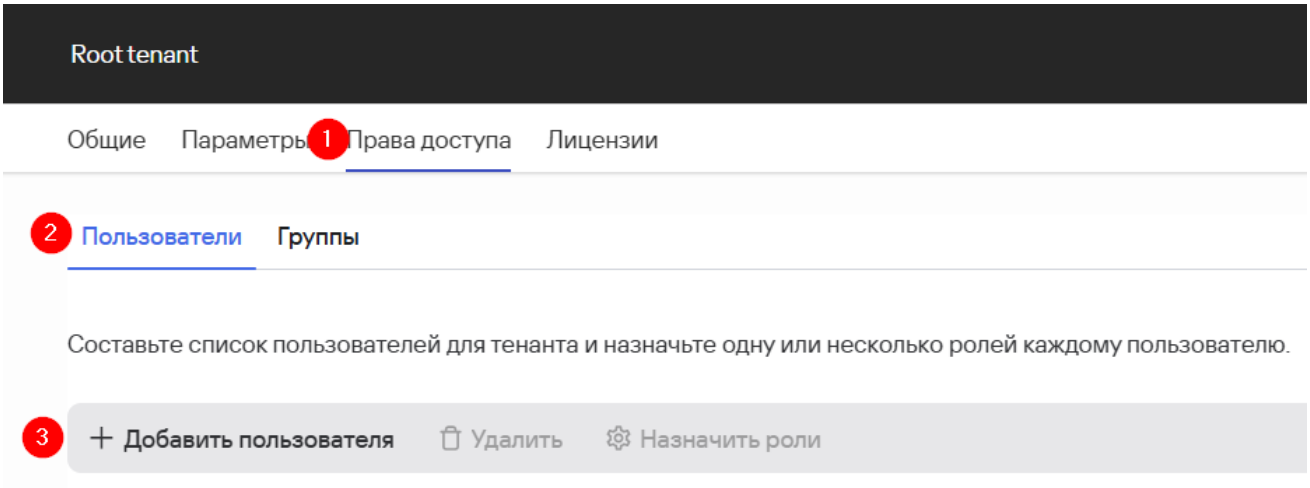


Пользователю назначена роль и теперь он может получить доступ к консоли SMP. Для получения доступа к функциям XDR необходимо дополнительно назначить роль пользователю в определенном тенанте.

4. Для этого перейдите в **Параметры** - **Тенанты** и нажмите на название необходимого тенанта для назначения пользователя

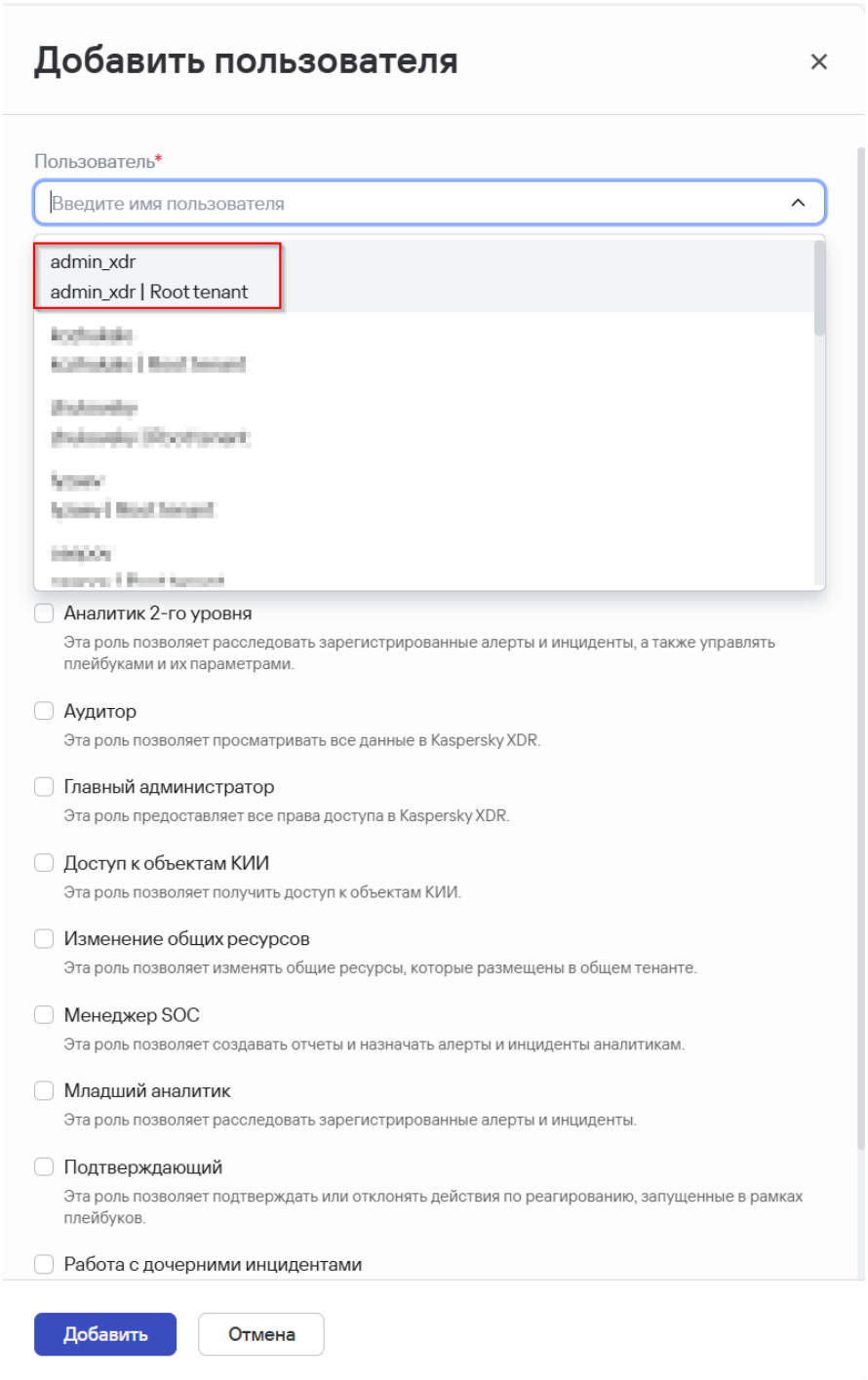


5. В открывшемся окне перейдите на вкладку **Права доступа**, в раздел **Пользователи** и нажмите на кнопку **Добавить пользователя**

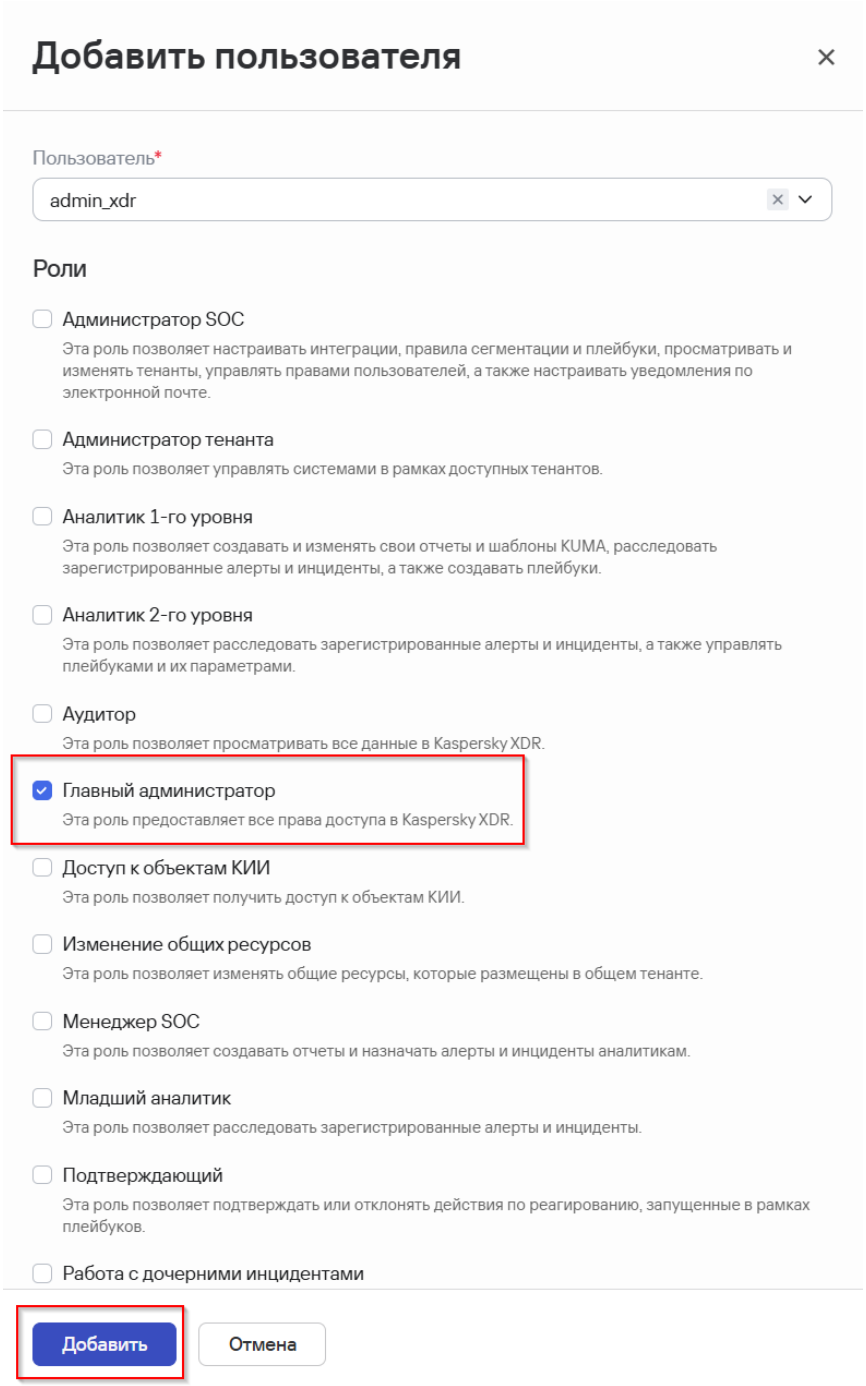


6. В выпадающем списке выберите нужного пользователя, которому собираетесь назначить роль

> Если нужного пользователя еще нет в списке, значит не прошел период синхронизации. Повторите попытку через несколько минут.



7. После выбора пользователя укажите выберите галочкой необходимые для него роли и нажмите кнопку **Добавить**



8. Не забудьте нажать кнопку **Сохранить** в таблице пользователей для применения изменений к Тенанту.

Готово. Пользователь создан, ему назначена роль для доступа SMP и роль доступа к функционалу XDR.

Замена сертификата консоли OSMP

Выпуск и замена сертификата консоли

Документ описывает выпуск серверного (листового) сертификата, сборку цепочки сертификатов и закрытого ключа в единый PEM-бандл и его установку в продукт (версии 2.1 и 2.2).

Процедура едина для двух вариантов инфраструктуры:

- **Вариант A** — доступен только корневой удостоверяющий центр (CA).
- **Вариант B** — доступны корневой и промежуточный CA.

Варианты отличаются значениями переменных в разделе 2. Остальные шаги идентичны.

---

1. Требования к бандлу

Итоговый файл — текстовый PEM, содержащий блоки в следующем порядке:

№	Содержимое
1	Листовой (серверный) сертификат
2	Промежуточные сертификаты — от нижнего к верхнему (при наличии)
3	Корневой сертификат
4	Закрытый ключ листового сертификата

Требования к закрытому ключу:

- без пароля;
- формат PKCS#8 — заголовок `-----BEGIN PRIVATE KEY-----`.

> Ключ с заголовком `-----BEGIN RSA PRIVATE KEY-----` (PKCS#1) продуктом не принимается. Команды раздела 4 создают ключ в требуемом формате; для преобразования существующего ключа см. раздел 8.5.

---

2. Подготовка

Исходные файлы CA должны находиться в рабочем каталоге в формате PEM. Если они предоставлены в другом формате, выполните раздел 8 и затем вернитесь сюда.

Задайте полное доменное имя сервиса:

```
DOMAIN="console.xdr.demo.lab"
```

Задайте переменные в соответствии с вариантом инфраструктуры.

Вариант A — только корневой CA

Требуемые файлы: `root.crt`, `root.key`.

```
ISSUING_CRT="root.crt"
ISSUING_KEY="root.key"
ROOT_CRT="root.crt"
CHAIN=""
```

Вариант B — корневой и промежуточный CA

Требуемые файлы: `root.crt`, `intermediate.crt`, `intermediate.key`.

```
ISSUING_CRT="intermediate.crt"
ISSUING_KEY="intermediate.key"
ROOT_CRT="root.crt"
CHAIN="intermediate.crt"
```

> При наличии нескольких промежуточных CA укажите их в `CHAIN` от нижнего к верхнему через пробел, а в `ISSUING_CRT` / `ISSUING_KEY` — самый нижний (непосредственно подписывающий листовой сертификат).

---

3. Генерация закрытого ключа и запроса на сертификат

```
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out console.key


openssl req -new -key console.key -out console.csr -subj "/CN=${DOMAIN}"
```

---

4. Подпись сертификата

```
openssl x509 -req -sha256 -days 365 \
    -CA "${ISSUING_CRT}" \
    -CAkey "${ISSUING_KEY}" \
    -set_serial 0x$(openssl rand -hex 16) \
    -in console.csr \
    -out console.crt \
    -extensions req_ext \
    -extfile <( \
      echo '[req_ext]'; \
      echo 'basicConstraints=CA:FALSE'; \
      echo 'keyUsage=digitalSignature,keyEncipherment'; \
      echo 'extendedKeyUsage=serverAuth'; \
      printf "subjectAltName=DNS:${DOMAIN}")
```

Для нескольких имён сервиса перечислите их в поле SAN через запятую, например:

```
printf "subjectAltName=DNS:${DOMAIN},DNS:console.smp.demo.lab,IP:10.0.0.10"
```

Проверка имени хоста выполняется по полю SAN, поэтому в нём должны быть указаны все адреса, по которым осуществляется доступ к сервису.

---

5. Сборка бандла

```
cat console.crt $CHAIN "${ROOT_CRT}" console.key > console_bundle.pem
```

Переменная `$CHAIN` указывается без кавычек.

Результат:

- Вариант A: `console.crt` → `root.crt` → `console.key`
- Вариант B: `console.crt` → `intermediate.crt` → `root.crt` → `console.key`

---

6. Проверка

6.1. Цепочка доверия

```
openssl verify -CAfile "${ROOT_CRT}" ${CHAIN:+-untrusted $CHAIN} console.crt
```

Ожидаемый результат: `console.crt: OK`.

6.2. Поле SAN

```
openssl x509 -in console.crt -noout -text | grep -A1 "Subject Alternative Name"
```

Убедитесь, что указано доменное имя сервиса.

6.3. Состав и формат бандла

```
grep "BEGIN" console_bundle.pem
```

Ожидаемый результат (Вариант B):

```
-----BEGIN CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
```

6.4. Соответствие ключа и сертификата

```
diff <(openssl x509 -in console.crt -noout -pubkey) <(openssl pkey -in console.key -pubout)
```

Пустой вывод подтверждает соответствие.

---

7. Установка бандла

Используйте команду в соответствии с версией продукта. `<путь к бандлу>` — путь к файлу `console_bundle.pem`.

Версия 2.1

```
./kdt invoke gateway -a updateCerts \
  --param console_bundle=<путь к бандлу> \
  --param bundle_list="console"
```

Версия 2.2

```
./kdt invoke gateway -a updateCertConsole \
  --param console_bundle=<путь к бандлу> \
  --param bundle_list="console"
```

---

8. Конвертация исходных файлов в PEM

Выполняется до раздела 2, если файлы CA предоставлены не в формате PEM.

Определение формата файла:

```
head -c 40 файл | grep -a "BEGIN" && echo "PEM" || echo "не PEM"
```

8.1. Сертификат в формате DER (.der, .cer)

```
openssl x509 -inform DER -in cert.der -out cert.pem
```

8.2. Ключ в формате DER

```
openssl pkey -inform DER -in key.der -out key.pem
```

8.3. Контейнер PKCS#12 (.pfx, .p12)

```
openssl pkcs12 -in bundle.pfx -clcerts -nokeys -out console.crt
openssl pkcs12 -in bundle.pfx -cacerts -nokeys -out ca-chain.pem
openssl pkcs12 -in bundle.pfx -nocerts -nodes  -out console.key
```

> При ошибке алгоритма в OpenSSL 3.x добавьте флаг `-legacy`. Файл `ca-chain.pem` при необходимости разделите на корневой и промежуточный сертификаты по блокам `BEGIN/END CERTIFICATE`.

8.4. Контейнер PKCS#7 (.p7b, .p7c)

```
openssl pkcs7 -print_certs -in chain.p7b -out chain.pem
```

8.5. Ключ PKCS#1 → PKCS#8

```
openssl pkcs8 -topk8 -nocrypt -in rsa.key -out console.key
```

8.6. Снятие пароля с ключа

```
openssl pkey -in encrypted.key -out console.key
```

---

9. Устранение неполадок

Браузер не устанавливает соединение, ошибка доверия к сертификату (возможно с упоминанием HSTS).

Корневой сертификат не является доверенным в браузере. Установите `root.crt` в доверенные корневые центры сертификации: в браузерах на базе системного хранилища

(Chrome, Edge, Safari) — в хранилище операционной системы; в Firefox — в собственном хранилище (Настройки → Приватность и защита → Сертификаты → Просмотр сертификатов → Центры сертификации → Импортировать, с отметкой «Доверять при идентификации веб-сайтов»). После установки перезагрузите страницу.

Ошибка сертификата во всех браузерах.

Доменное имя не совпадает со значением поля SAN. Перевыпустите сертификат (разделы 3–5) с корректным значением `DOMAIN` либо добавьте требуемые имена в SAN (раздел 4).

Ключ не принимается продуктом.

Проверьте заголовок ключа (раздел 6.3). Требуется `-----BEGIN PRIVATE KEY-----`. При заголовке `-----BEGIN RSA PRIVATE KEY-----` выполните конвертацию (раздел 8.5) и пересоберите бандл.

Ошибка `unable to get local issuer certificate` при проверке.

В бандле отсутствует промежуточный или корневой сертификат либо нарушен порядок блоков. Сверьтесь с разделом 1 и пересоберите бандл.