vscalecommunity

Настройка HTTPS в GitLab с помощью Let's Encrypt

Vscale
11 декабрь 2017 г. 21:37

Для получения SSL сертификата мы воспользуемся бесплатным сервисом Let's Encrypt. Для выполнения этой части руководства необходимо быть владельцем доменного имени, которое можно получить у любого интернет-регистратора. Далее мы будем использовать **example.ru** и его нужно будет заменять на существующее доменное имя при следовании данному руководству.

Изменение DNS-серверов для доменного имени

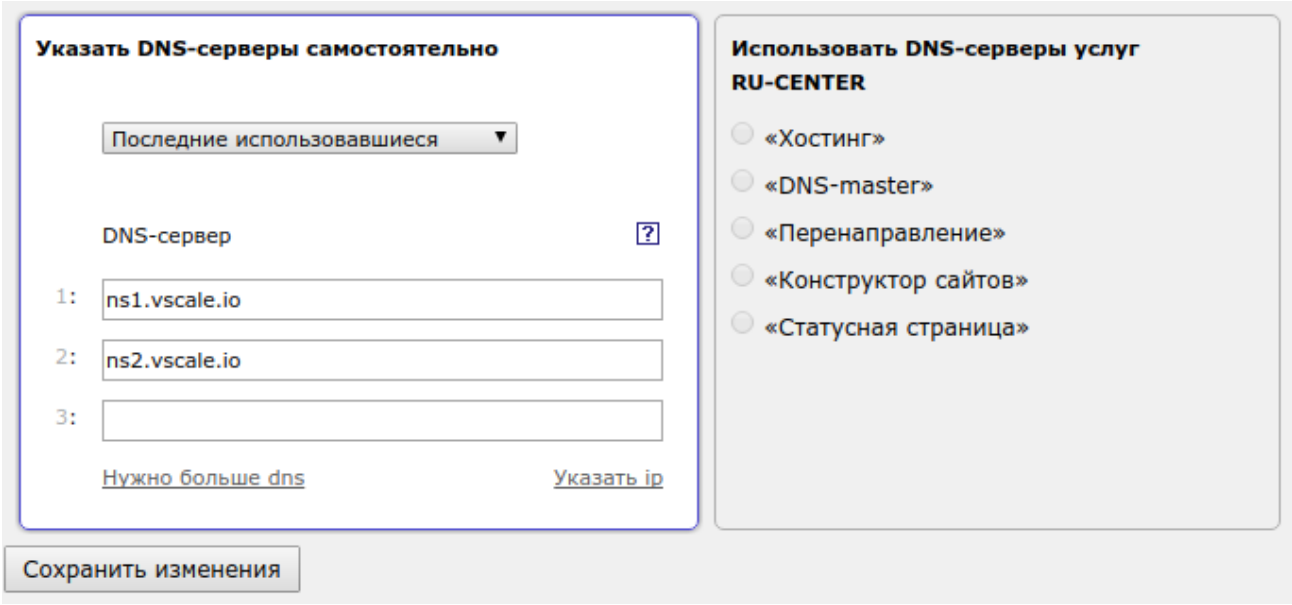
Перед добавлением доменного имени в Vscale необходимо перейти в панель управления Вашего интернет-регистратора и указать для выбранного домена ns-записи:

- ns1.vscale.io
- ns2.vscale.io

Общий алгоритм одинаков для большинстве регистраторов:

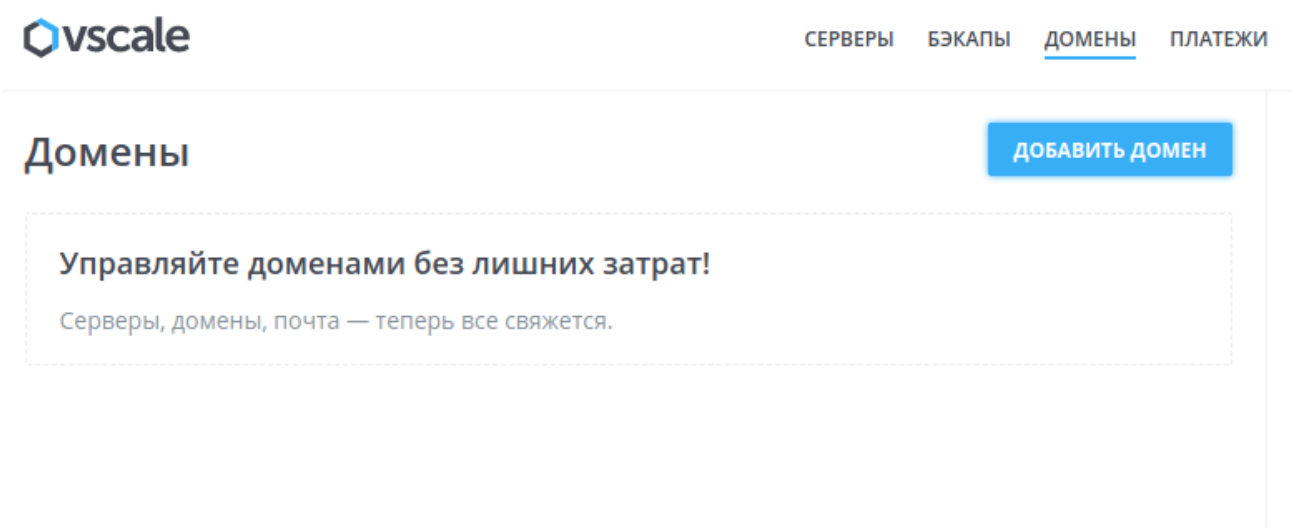
1. Перейти в управление доменами;
2. Выбрать определенный домен;
3. Выбрать управление DNS-серверами;
4. Удалить существующие записи;
5. Прописать новые ns-записи, указанные выше.

Например, управление DNS-серверами на сайте компании "RU-CENTER":

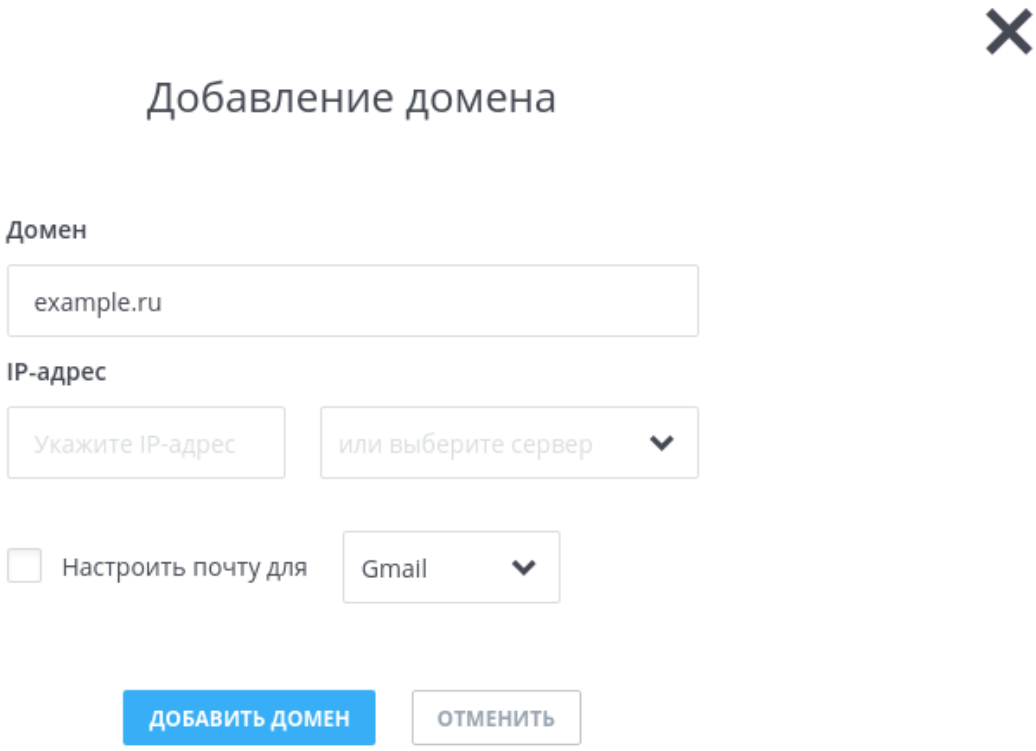


Добавление доменного имени в панель Vscale

В панели управления Vscale в разделе **Домены** добавим новый домен:



Укажем доменное имя, а IP-адрес пока оставим пустым:



Теперь создадим поддомен **gitlab** и свяжем его с IP-адресом сервера, на котором запущен GitLab:

Потребуется некоторое время до того, как GitLab станет доступен по адресу :

`http://gitlab.example.ru`

Подключимся по SSH к серверу GitLab и дальнейшие действия будем выполнять в нём.

Прежде чем мы получим SSL сертификат для нашего GitLab приложения, необходимо установить Certbot - официальный Let's Encrypt клиент.

Для начала добавим репозиторий Certbot от разработчиков:

```
$ apt install software-properties-common
$ add-apt-repository ppa:certbot/certbot
```

Теперь обновим список программных пакетов и установим Certbot и текстовый редактор vim:

```
$ apt-get update
$ apt-get install certbot vim
```

Подготовка к подтверждению владения доменом

Для получения SSL-сертификата от удостоверяющего центра Let's Encrypt необходимо подтвердить владение доменом.

Текущая установка GitLab через Omnibus включает в себя Nginx для внутреннего обслуживания запросов, поэтому наилучшим способом подтверждения домена является метод Webroot. Он заключается в использовании существующего веб-сервера, в нашем случае Nginx, для предоставления специального файла по пути **/.well-known** через 80 порт.

Создадим пустую директорию для пользования Certbot-ом:

```
$ mkdir -p /var/www/letsencrypt
```

Теперь откроем конфигурационный файл GitLab для настройки Nginx:

```
$ vim /etc/gitlab/gitlab.rb
```

Внутри файла находятся конфигурации компонентов и сервисов, которые использует GitLab. Перейдем в раздел Nginx для связывания пути **/.well-known** с созданной директорией. Для этого введем **:826** и нажмем клавишу **Enter.**

Теперь нажмем клавишу **i** для входа в режим редактирования и вставим следующую строчку:

```
nginx['custom_gitlab_server_config'] = "location ^~ /.well-known { root /var/
```

Для выхода из режима редактирования нажмите клавишу **Esc**. Для сохранения изменений и выхода из файла введите команду **:wq** и нажмите клавишу **Enter**.

Перезагрузим сервис для вступления изменений в силу:

```
$ gitlab-ctl reconfigure
```

Получение сертификата через Certbot

Поскольку автоматическая настройка веб-сервера Certbot-ом невозможна, мы запросим только сертификат через команду **certonly** и укажем директорию для проверки файла, а также доменное имя (не забудьте поменять gitlab.example.ru на настоящий адрес):

```
$ certbot certonly --webroot --webroot-path=/var/www/letsencrypt -d gitlab.e
```

Certbot во время регистрации запросит email адрес. Важно указать действующий адрес для получения сообщений об оставшемся сроке действия сертификата. После подтверждения валидности домена мы получим такое сообщение:

```
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/gitlab.example.ru/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/gitlab.example.ru/privkey.pem
   Your cert will expire on 2018-02-18. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot
   again. To non-interactively renew *all* of your certificates, run
   "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
```

Certbot информирует о сроке действия сертификата (3 месяца), а также способах продления его в будущем.

Теперь настроим Nginx для использования сертификата. Откроем конфигурационный файл:

```
$ vim /etc/gitlab/gitlab.rb
```

Перейдем в конец файла, набрав команду **:$** и нажав клавишу **Enter**

Войдем в режим редактирования, нажав клавишу **i** и удалим запись:

```
external_url 'http://localhost'
```

Выйдем из режима редактирования нажатием клавиши **Esc**.

Теперь вернемся в начало файла командой **:1** и нажатием **Enter**. Снова войдем в режим редактирования **i** и поменяем строчку:

```
external_url 'http://gitlab.example.com'
```

на

```
external_url 'https://gitlab.your_domain.ru'
```

Выйдем из режима редактирования клавишей **Esc**.

Осталось настроить Nginx для работы с полученными сертификатами. Перейдем в раздел Nginx командой **:826** и войдём в режим редактирования, нажав клавишу **i**. Теперь вставим следующий текст, не забыв поменять **example.ru** на свой домен:

```
nginx['redirect_http_to_https'] = true
nginx['ssl_certificate'] = "/etc/letsencrypt/live/gitlab.example.ru/fullchai
nginx['ssl_certificate_key'] = "/etc/letsencrypt/live/gitlab.example.ru/priv
```

Выйдем из режима редактирования, нажав клавишу **Esc** и сохраним изменения набрав **:wq** и нажав клавишу **Enter**.

Перезапустим сервис для сохранения изменений:

```
$ gitlab-ctl reconfigure
```

Для проверки установки перейдем по адресу в браузере:

http://gitlab.example.ru

Если всё прошло успешно, то произойдет перенаправление на https и статус сайта поменяется на "Надёжный".

Заключение

Мы обезопасили наше приложение GitLab и теперь можно приступать к более сложным вещам, например, к настройке непрерывной интеграции через GitLab CI.

Стоит отметить, что полученный нами сертификат валиден всего 3 месяца, поэтому в будущем рекомендуем создать задание Cron для автоматического продления сертификата.

1

Комментарии

1 комментарий Сортировка

dyp2000
26 сентябрь 2018 г. 18:42

0

Спасибо за статью. Очень помогла.

Войдите в службу, чтобы оставить комментарий.

Не нашли то, что искали?

Задать вопрос