

Достаточно долгое время у меня не получалось корректно настроить одновременную работу Gitlab и встроенный Container Registry. Сложность состояла в том, что все это хозяйство работало за внешним nginx reverse proxy с настроенным https. С прокси же запросы на сам gitlab шли по http и на сервере сертификатов не было вовсе. Это важно, так как не хотелось одновременно настраивать и обновлять сертификаты на nginx proxy и gitlab.

Онлайн-курс по Kubernetes – для разработчиков, администраторов, технических лидеров, которые хотят изучить современную платформу для микросервисов Kubernetes. Самый полный русскоязычный курс по очень востребованным и хорошо оплачиваемым навыкам. Курс не для новичков – нужно пройти .

Содержание

Введение

Gitlab и проху nginx

[Настройка Gitlab registry за reverse nginx proxy](#)
[Проверка работы gitlab registry](#)
[Часто задаваемые вопросы по теме статьи \(FAQ\)](#)
[Заключение](#)
[Помогла статья? Подписывайся на telegram канал автора](#)

Введение

Просто настроить gitlab за nginx в режиме proxy_pass не представляет никакой сложности. Трудности возникают именно тогда, когда вы хотите использовать встроенный Container Registry. Я несколько раз подходил к этой задаче и каждый раз отступал, либо отказываясь от registry, либо от nginx proxy.

Без прокси неудобно, так как чаще всего, если у вас есть какая-то инфраструктура, а не одиночный сервер, смотреть напрямую в интернет у вас будет какой-то шлюз, а не сам сервис. Так удобно разграничивать доступ, контролировать и перенаправлять подключения, следить за сертификатами, балансировать нагрузку и т.д. В общем, я использую его всегда, когда это возможно и оправдано (почти всегда).

С registry не получалось корректно настроить работу за прокси. Я перепробовал все советы, что только нашел в интернете. Перечитал всю документацию, но так ничего не получилось. Команда docker login успешно обрабатывала на сервере, а вот запустить образ в реджистри gitlab никак не получалось. Получал ошибку.

gitlab. В качестве **EXTERNAL_URL** во время установки можно указать адрес с http. Затем, после установки, настраивайте проху nginx примерно вот так.

```
server {
    listen 443 http2 ssl;
    server_name gl.serveradmin.ru;
    access_log /var/log/nginx/gl.serveradmin.ru-access.log full;
    error_log /var/log/nginx/gl.serveradmin.ru-error.log;

    ssl_certificate /etc/letsencrypt/live/gl.serveradmin.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/gl.serveradmin.ru/privkey.pem;

    limit_conn perip 50;

    location /.well-known {
        root /tmp;
    }

    location / {
        proxy_pass http://10.20.50.8:80;
        proxy_read_timeout    300;
        proxy_connect_timeout  300;
        proxy_redirect         off;
        proxy_set_header       X-Forwarded-Proto https;
        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-Ssl on;
    }
}

server {
    listen 80;
```

```
server_name gl.serveradmin.ru;  
return 301 https://gl.serveradmin.ru$request_uri;  
}
```

На самом сервере с gitlab в `/etc/gitlab/gitlab.rb` измените параметры.

```
external_url 'https://gl.serveradmin.ru'  
nginx['listen_port'] = 80  
nginx['listen_https'] = false
```

И после этого дайте команду на переконфигурацию системы.

```
gitlab-ctl reconfigure
```

После этого gitlab будет нормально работать за nginx проху. Дальше рассказываю, что надо сделать, чтобы включить и нормально запустить в работу Container Registry за reverse proxy.

Настройка Gitlab registry за reverse nginx проху

На первый взгляд никаких особенных сложностей быть не должно. Делаем по аналогии настройки, только для registry_nginx и пробуем работать с registry. Но ничего не получится. Будет приведенная выше ошибка. Я покопался в настройках gitlab и понял, в чем проблема.

Если у вас используется одинаковый url для web интерфейса gitlab и registry, то корректная работа за nginx проху невозможна. Я посмотрел конфигурацию nginx, входящего в состав gitlab. Она живет тут - `/var/opt/gitlab/nginx/conf`. Там отдельный конфиг для web интерфейса и для registry. Если используются одинаковые доменные имена, то все запросы попадают на конфигурацию веб интерфейса и проксируются на gitlab-workhorse.

Это происходит, потому что директива `server_name` в обоих конфигах одинаковая. Обрабатывается первым то, что стоит раньше в конфигурации. Я заподозрил это еще раньше, когда 404 ошибки обращений к registry обнаруживал в лог файлах `/var/log/gitlab/nginx/gitlab_access.log`, тогда как

`gitlab_registry_access.log` были пустые.

Сбивало с толку то, что авторизация docker login при этом обрабатывала корректно, а ошибка возникала только во время push. Судя по всему, процесс авторизации один и тот же, что для registry, что для http доступа. В общем, чтобы все корректно работало, вам нужно разделить на разные домены gitlab и registry.

Я сделал для registry отдельный домен. Настроил для него проксирование, примерно так.

```
server {
    listen 443 http2 ssl;
    server_name rg.serveradmin.ru;
    access_log /var/log/nginx/rg.serveradmin.ru-access.log full;
    error_log /var/log/nginx/rg.serveradmin.ru-error.log;

    ssl_certificate /etc/letsencrypt/live/rg.serveradmin.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/rg.serveradmin.ru/privkey.pem;

    limit_conn perip 50;

    location /.well-known {
        root /tmp;
    }

    location / {
        proxy_pass http://10.20.50.8:80;
        proxy_read_timeout    300;
        proxy_connect_timeout  300;
        proxy_redirect        off;
        proxy_set_header      X-Forwarded-Proto https;
        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-Ssl    on;
proxy_set_header    X-Frame-Options    SAMEORIGIN;
proxy_cache off;
proxy_buffering off;
proxy_request_buffering off;
proxy_http_version 1.1;
}
}

server {
    listen 80;
    server_name rg.serveradmin.ru;
    return 301 https://rg.serveradmin.ru$request_uri;
}
```

Дальше в *gitlab.rb* добавил параметры, отвечающие за работу registry.

```
registry_external_url 'https://rg.serveradmin.ru'
gitlab_rails['registry_enabled'] = true
registry['enable'] = true
registry_nginx['enable'] = true
registry_nginx['proxy_set_headers'] = {
  "Host" => "$http_host",
  "X-Real-IP" => "$remote_addr",
  "X-Forwarded-For" => "$proxy_add_x_forwarded_for",
  "X-Forwarded-Proto" => "https",
  "X-Forwarded-Ssl" => "on"
}
registry_nginx['listen_port'] = 80
```

```
registry_nginx['listen_https'] = false
```

Я не уверен, что нужны все эти параметры. Привожу пример с реально работающего сервера и чтобы случайно чего-то не забыть, привожу сразу все, что на нем сейчас активно из настроек. Думаю, заголовки указывать не обязательно. Без них тоже заработает.

После этого перечитывайте конфигурацию gitlab и nginx и проверяйте работу.

Проверка работы gitlab registry

Теперь проверим работу Gitlab Container Registry. Для этого создаем любой проект через web интерфейс. Идем в раздел **Packages -> Container Registry**. Там увидите краткую инструкцию по работе с реджистри.

The screenshot shows the GitLab Container Registry interface for a project named 'myapp'. The left sidebar contains navigation options: Project overview, Repository, Issues (0), Merge Requests (0), CI / CD, Operations, Packages (highlighted with a red box), Container Registry, Wiki, Snippets, and Settings. The main content area displays a message: 'There are no container images stored for this project' with a ship icon. Below this, it explains that every project can have its own space for Docker images and provides a 'Quick Start' section. The 'Quick Start' section includes instructions on authentication and a code block containing three Docker commands: 'docker login rg.serveradmin.ru', 'docker build -t rg.serveradmin.ru/zeroxzed/...', and 'docker push rg.serveradmin.ru/zeroxzed/myapp'. The code block is highlighted with a red box.

Идем на сервер, где будем собирать образы docker. Настраиваем подключение к registry, указав логин с паролем.

```
docker login rg.serveradmin.ru
Username: zeroxzed
Password:
```

```
Login Succeeded
```

Создадим тестовый докер образ из Dockerfile.

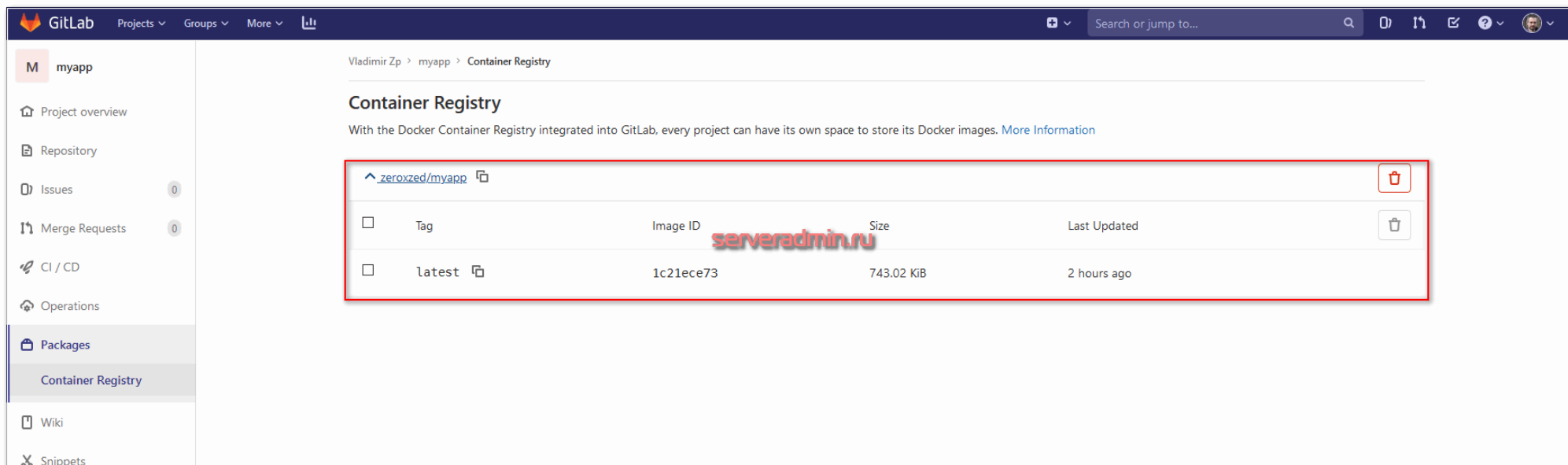
```
FROM busybox  
RUN echo "Test OK"
```

Собираем образ и загружаем его в registry.

```
# docker image build -t rg.serveradmin.ru/zeroxzed/myapp:latest .  
# docker push rg.serveradmin.ru/zeroxzed/myapp:latest
```

```
root@ubuntu18:~/myproject# docker image build -t rg.serveradmin.ru/zeroxzed/myapp:latest .
Sending build context to Docker daemon 3.072kB
Step 1/2 : FROM busybox
--> be5888e67be6
Step 2/2 : RUN echo "Test OK"
--> Using cache
--> 1c21ece73128
Successfully built 1c21ece73128
Successfully tagged rg.serveradmin.ru/zeroxzed/myapp:latest
root@ubuntu18:~/myproject# docker push rg.serveradmin.ru/zeroxzed/myapp:latest
The push refers to a repository [rg.serveradmin.ru/zeroxzed/myapp]
5b0d2d635df8: Layer already exists
latest: digest: sha256:3e7198f651217f2435f111fcae0ac5135c29117e49a4d166813597e85623e91b size: 527
root@ubuntu18:~/myproject#
```

Указанный образ будет загружен в Registry данного проекта.



Часто задаваемые вопросы по теме статьи (FAQ)

Почему запросы с проху на сервер gitlab идут без шифрования по протоколу http?

В моем примере запросы с прокси на рабочий сервер идут в рамках закрытой непубличной инфраструктуры, где вероятность перехвата трафика минимальна. Нет смысла тратить ресурсы и время на настройку и обновления сертификатов не только на nginx проху, но и рабочем сервере.

Возможно ли настроить подобное проксирование со шлюза на рабочий сервер через интернет?

Да, это возможно. Принципиальной разницы нет. Единственное отличие - нужно использовать протокол https при перенаправлении запросов с прокси на сервер gitlab.

Можно ли таким же способом проксировать запросы к встроенным gitlab pages или mattermost?

Да, настраивается это примерно так же. Необходимо создать поддомен для каждого сервиса и по аналогии с приведенным примером настроить

перенаправление запросов с nginx на gitlab.

У вас есть подробный материал, где рассказано о том, как настроить и начать работать с gitlab?

Да, у меня есть отдельная статья - установка, настройка и работа с gitlab.

Заключение

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении. Расскажи, как сделать правильно!

На этом все по работе gitlab за nginx proxy. Настройка очень простая, продуктом пользоваться удобно. Настроить типовой ci/cd на базе докера очень просто. В будущем опишу такой пример. Так что рекомендую Gitlab для совместной работы команды. Минус у него только один - очень требователен к ресурсам.

Онлайн курс по Kubernetes

Онлайн-курс по Kubernetes - для разработчиков, администраторов, технических лидеров, которые хотят изучить современную платформу для микросервисов Kubernetes. Самый полный русскоязычный курс по очень востребованным и хорошо оплачиваемым навыкам. Курс не для новичков - нужно пройти вступительный тест.

Если вы ответите "да" хотя бы на один вопрос, то это ваш курс:

- устали тратить время на автоматизацию?
- хотите единообразные окружения?;
- хотите развиваться и использовать современные инструменты?
- небезразлична надежность инфраструктуры?
- приходится масштабировать инфраструктуру под растущие потребности бизнеса?

- хотите освободить продуктовые команды от части задач администрирования и автоматизации и сфокусировать их на развитии продукта?

Сдавайте вступительный тест по и присоединяйтесь к новому набору!.

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.