

Пояснение по учётному времени на перенос GitLab

Коллеги, подробно поясняю ситуацию с переносом GitLab и состав работ, из которых фактически сформировалось учётное время.

1. Перенос не сводился к загрузке архива

Речь не шла о том, что небольшой архив репозитория передавался в течение восьми часов.

Размер репозитория составлял около 47 МБ. При наличии корректно подготовленных веток, открытых сетевых доступов и понятной схемы окружений физическое клонирование или зеркальный перенос такого репозитория действительно занимает несколько минут.

Основное время было затрачено на три группы работ:

- Определение фактического состояния PROD- и DEV-окружений.
- Сверку и выравнивание веток старого и нового репозитория.
- Диагностику инфраструктурных ограничений при переключении серверов на внутренний GitLab.

2. Определение фактического состояния PROD и DEV

На момент начала работ отсутствовала однозначная и подтверждённая схема, при которой:

- ветка `main` полностью соответствовала PROD;
- ветка `develop` полностью соответствовала DEV.

PROD был развёрнут из локального состояния одной из рабочих веток. После развёртывания эта же ветка продолжала изменяться и использоваться в дальнейшей разработке.

В результате ветка с одинаковым названием на PROD, DEV и в GitLab могла указывать на разные коммиты и содержать различающийся код.

С DEV ситуация также была неоднозначной: рабочая ветка в разное время фигурировала под несколькими названиями. Поэтому выполнить обычное зеркальное копирование репозитория и считать перенос завершённым было невозможно.

Сначала требовалось определить, какой код фактически работает на каждом сервере, и зафиксировать контрольные состояния окружений.

3. Проверка и выравнивание PROD

Для PROD была создана отдельная резервная ветка, зафиксировавшая код, который фактически находился на сервере.

После этого новый `main` сравнивался с контрольным состоянием PROD.

В ходе проверки анализировались истории веток, включавшие более 600 коммитов. Были обнаружены дополнительные коммиты и фактические различия в файлах.

Отдельно проверялись изменения в:

- backend;
- Next.js;
- lock-файлах;
- `.gitignore`;
- конфигурациях окружений.

Конфигурации DEV нельзя было механически переносить в PROD, поэтому различия требовали ручной проверки.

После выполнения сверки и выравнивания было подтверждено, что новый `main` соответствует контрольному состоянию PROD.

4. Повторное восстановление и перенос DEV

Контрольная проверка DEV показала, что первоначально перенесённая ветка не являлась точной копией кода, фактически развёрнутого на стенде.

Ветки имели общую базу, но отличались верхними коммитами и историей слияний.

Поэтому потребовалось повторно выполнить следующие действия:

- определить фактическое состояние DEV;
- восстановить ветку из состояния сервера;
- отправить восстановленную ветку в новый GitLab;
- сравнить её со старым репозиторием;
- подготовить и проверить Merge Request;
- привести новый `develop` в соответствие с реально работающим DEV;
- повторно собрать и проверить стенд.

5. Инфраструктурные ограничения на DEV

После подготовки репозитория возник отдельный инфраструктурный blocker при непосредственном переключении серверов на новый GitLab.

29 июня DEV-сервер не мог установить соединение с внутренним GitLab:

- SSH, порт 22 — timeout;
- альтернативный SSH, порт 2222 — timeout;
- HTTPS, порт 443 — timeout;
- команды `git fetch` и `git ls-remote` зависали.

Для определения причины выполнялись проверки:

- DNS;

- маршрутизации;
- исходящего IP-адреса;
- доступности портов;
- настроек Git remote.

Диагностика показала, что DEV-сервер физически не может подключиться к GitLab из-за сетевых ограничений.

Было установлено, что на firewall GitLab необходимо разрешить доступ с исходящего IP-адреса DEV-сервера.

Изменение требовалось выполнить на стороне инфраструктуры клиента, поскольку у нашей команды отсутствовал доступ к firewall.

30 июня было подтверждено, что ограничение ещё не устранено. До изменения сетевых правил продолжить переключение было технически невозможно.

1 июля после получения информации об исправлении доступ был проверен повторно. Изначально соединение по-прежнему не работало, поэтому был найден обходной вариант подключения через HTTPS вместо SSH.

После этого DEV был:

- переключён на новый репозиторий;
- повторно собран;
- протестирован.

Корректность работы DEV была подтверждена.

6. Инфраструктурные ограничения на PROD

Аналогичное ограничение отдельно проявилось на PROD.

У PROD-сервера был другой исходящий IP-адрес, поэтому разрешение, выполненное для DEV, не обеспечивало доступ с PROD.

На PROD также не работали подключения по SSH и HTTPS, а Git-команды зависали при попытке обращения к репозиторию.

Дополнительно была обнаружена проблема с DNS на самой виртуальной машине.

После передачи информации инфраструктурной команде и корректировки DNS соединение было восстановлено.

Затем были выполнены:

- переключение Git remote;
- проверка ветки `main`;
- слияние Merge Request;
- сборка и запуск PROD;
- финальная проверка окружения.

1 июля в 14:16 работа PROD из нового GitLab была подтверждена.

7. Фактически выполненные работы

В рамках задачи были выполнены:

- аудит старого и нового репозитория;
- определение реального состояния PROD и DEV;
- создание контрольных серверных веток;
- сверка истории и содержимого веток;
- выравнивание `main` с PROD;
- обнаружение некорректно перенесённого `develop`;
- повторное восстановление и перенос DEV-ветки;
- работа с Merge Requests и защищёнными ветками;
- диагностика недоступности GitLab с DEV и PROD;
- проверка DNS, маршрутизации и портов 22, 2222 и 443;
- подготовка технической информации для инфраструктурной команды;
- повторные проверки после внесения изменений;
- корректировка DNS на PROD-VM;
- переключение Git remote на обоих серверах;
- сборка и проверка DEV;
- сборка и проверка PROD.

8. Почему сравнение с обычным зеркальным переносом не отражает ситуацию

Сравнение данной задачи с переносом нескольких десятков микросервисов за 30 минут не учитывает исходные условия.

Если репозитории заранее подготовлены, ветки соответствуют окружениям, доступы открыты, а задача сводится исключительно к выполнению:

```
clone --mirror
```

и

```
push --mirror,
```

такой перенос действительно выполняется быстро.

В данном случае задача заключалась не в передаче 47 МБ Git-объектов, а в:

- восстановлении фактического состояния двух окружений;
- выравнивании веток;
- проверке расхождений;
- восстановлении некорректно перенесённого DEV;

- работе с Merge Requests;
- переключении защищённых серверов на внутренний GitLab;
- устранении сетевых и DNS-ограничений, первоначально блокировавших доступ к репозиторию.

9. Итог

Понимаем, что цифра в 8 часов для задачи, которая на первый взгляд выглядит как «просто переключить гит», может вызывать вопросы — это объяснимая реакция, если не видеть весь объём работ изнутри.

Как видно из разделов выше, в эту цифру вошли: приведение веток в соответствие с фактическим состоянием PROD и DEV, повторная работа с веткой DEV после обнаруженного расхождения, а также диагностика и устранение сетевых ограничений при подключении к GitLab — отдельно для DEV и отдельно для PROD.

Время ожидания ответа от инфраструктурной команды между обращениями (в частности, с вечера 29 июня до утра 1 июля) в учётные часы не включалось — в этот период продолжить работу с нашей стороны было технически невозможно без изменений на стороне сервера.