

Онлайн образование

otus.ru



Проверить, идет ли запись

Меня хорошо видно && слышно?



Тема вебинара

Git



Лавлинский Николай

Технический директор “Метод Лаб”

<https://www.methodlab.ru/>

<https://www.youtube.com/c/NickLavlinsky>

<https://vk.com/nick.lavlinsky>



Преподаватель



Лавлинский Николай

Более 15 лет в веб-разработке

Преподавал в ВУЗе более 10 лет
Более 3 лет в онлайн-образовании

Специализация: оптимизация производительности,
ускорение сайтов и веб-приложений

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Slack #general



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом

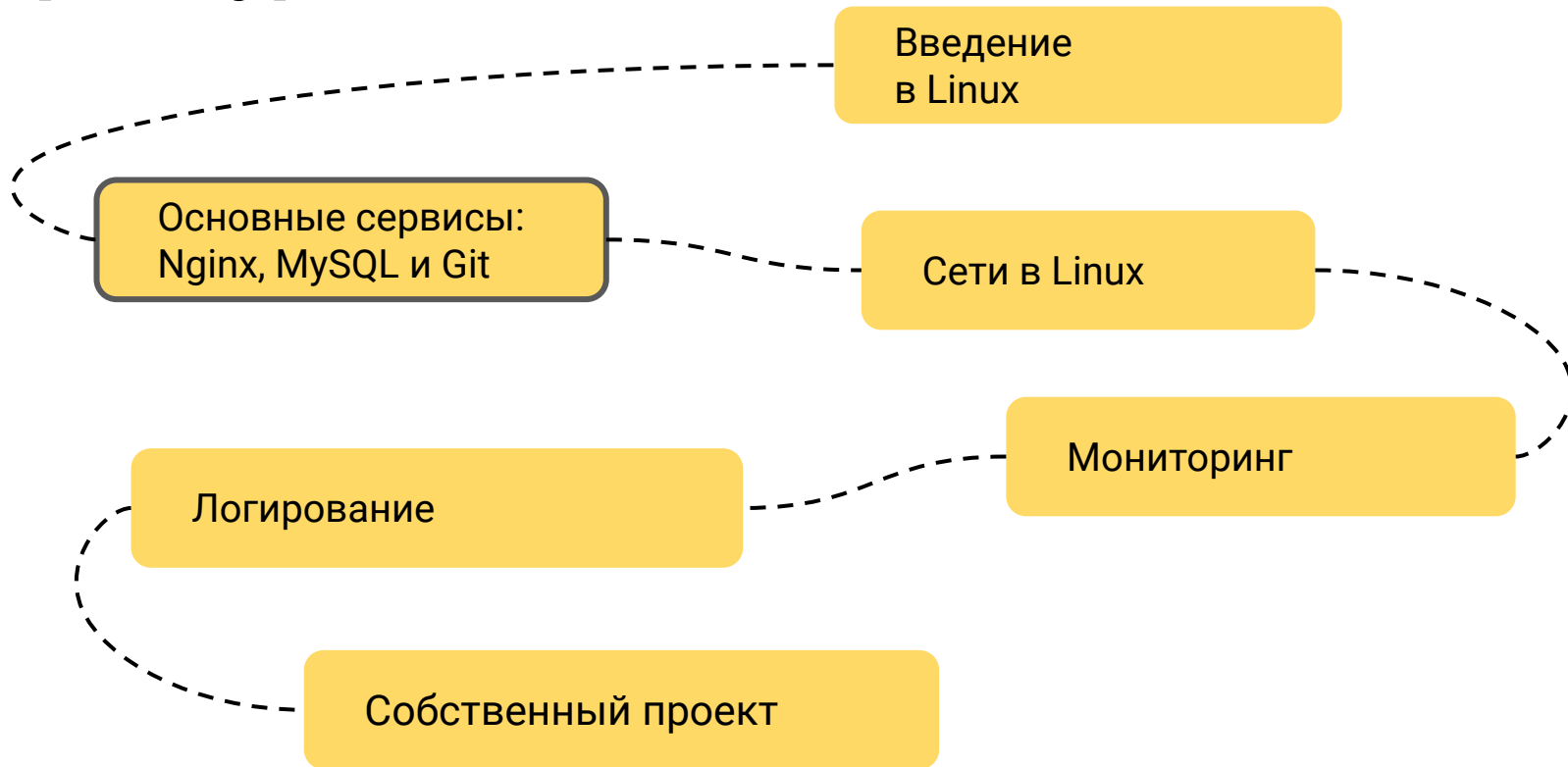


Документ



Ответьте себе или
задайте вопрос

Карта курса



Маршрут вебинара



Управление версиями

Основы Git

Работа с Git с CLI

Удалённые репозитории GitHub

Цели вебинара

После занятия вы сможете

1. Понимать основные термины в Git
2. Получать исходный код из репозитория
3. Проводить основные манипуляции с файлами
4. Создавать удалённые репозитории в GitHub

Смысл

Зачем вам это уметь

1. Получать код приложений
2. Управлять конфигурацией в Git
3. Вносить изменения в код
4. Автоматизация доставки приложений



Системы контроля версий

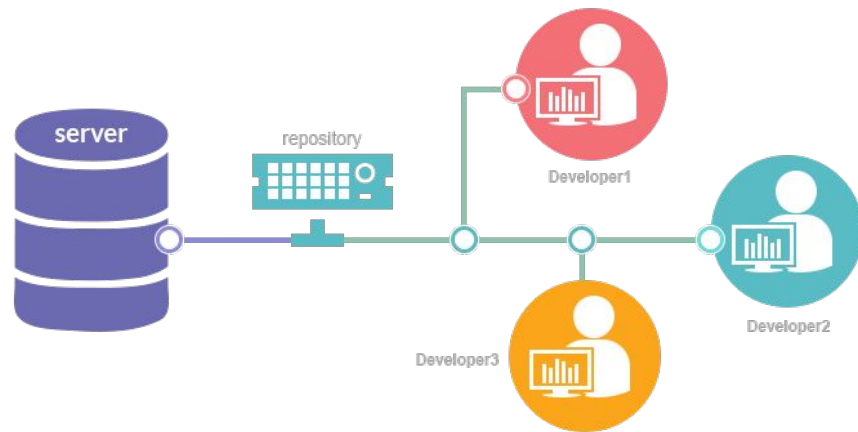


Что такое, зачем?

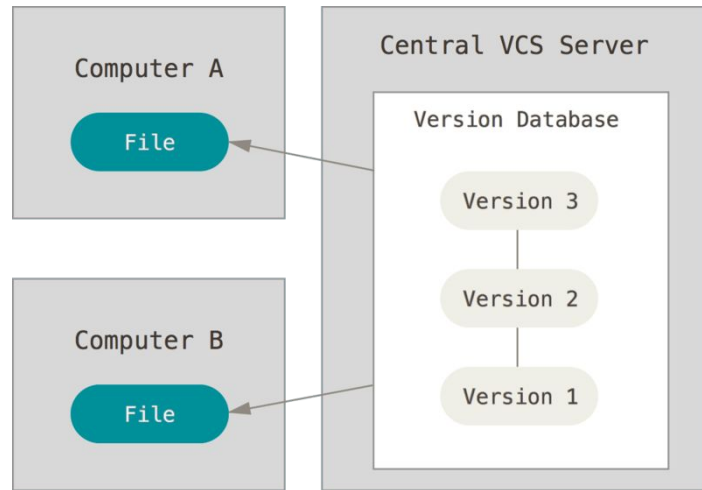
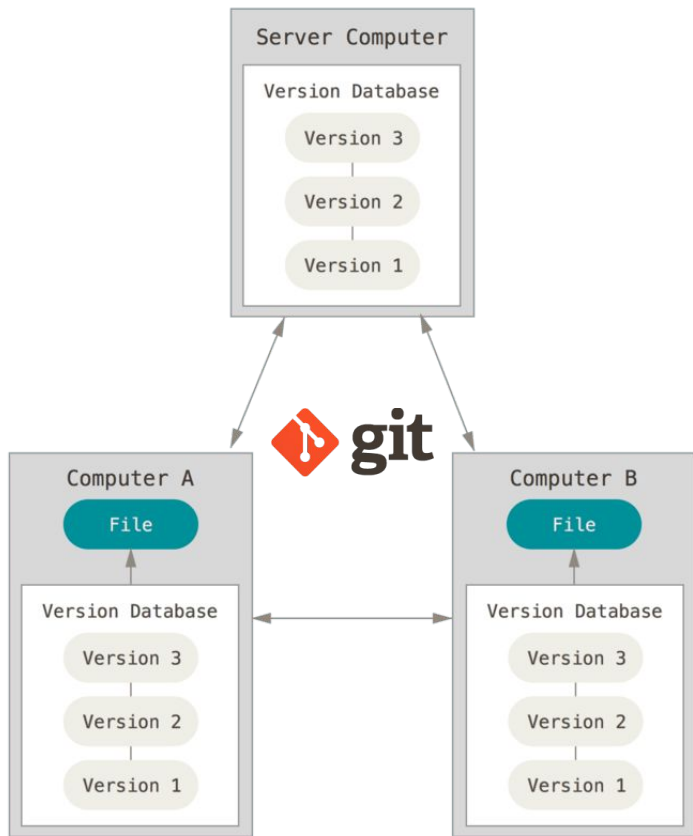


Задачи version control system (VCS)

- Учет версий софта
- Резервные копии, откат
- Параллельная разработка
- Автоматизация сборки, доставки
- Документация изменений



Распределённые и централизованные



Преимущества децентрализованных

- Полная локальная копия
- Локальный проект это бекап
- Возможность работы оффлайн
- Низкая нагрузка на инфраструктуру

Базовые термины

- Репозиторий (repository, repo)
- Рабочий каталог (working directory)
- Коммит (commit)
- Область подготовленных файлов (staged area)

Создание репозитория

```
git init

# Создаём каталог
mkdir repo
cd repo

# Создаём репозиторий
git init

# Проверяем
git status
```



Базовые настройки

git config



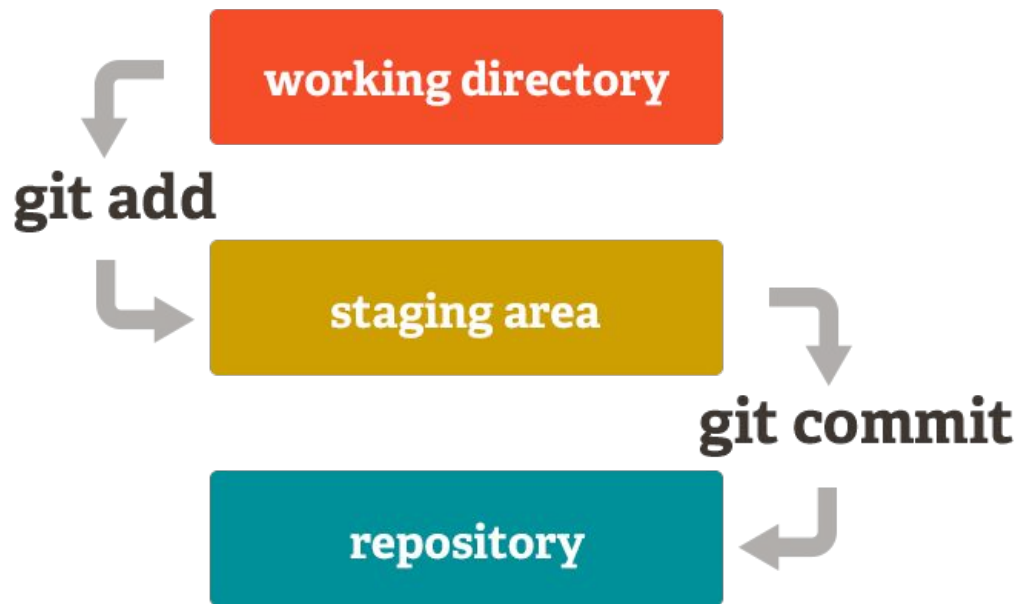
```
# Свойства пользователя  
git config --global user.name "first_name last_name"  
git config --global user.email email@domain  
  
# Редактор (необязательно)  
git config --global core.editor nano
```



Содержимое каталога .git

- Конфигурация
- Объекты (Commit, Tree, Blob)
- Ссылки на коммиты и ветки
- Скрипты-хуки

Область подготовленных файлов



Добавление в индекс и удаление

git add remove



Добавление в индекс (stage)

```
git add testfile
```

Просмотр проиндексированных изменений

```
git diff --cached
```

Удаление из индекса (stage)

```
git rm --cached testfile
```

Удаление файла (совсем)

```
git rm testfile
```



Создаём коммит

git commit



Добавляем файлы

```
git add <filename/wildcard>
```

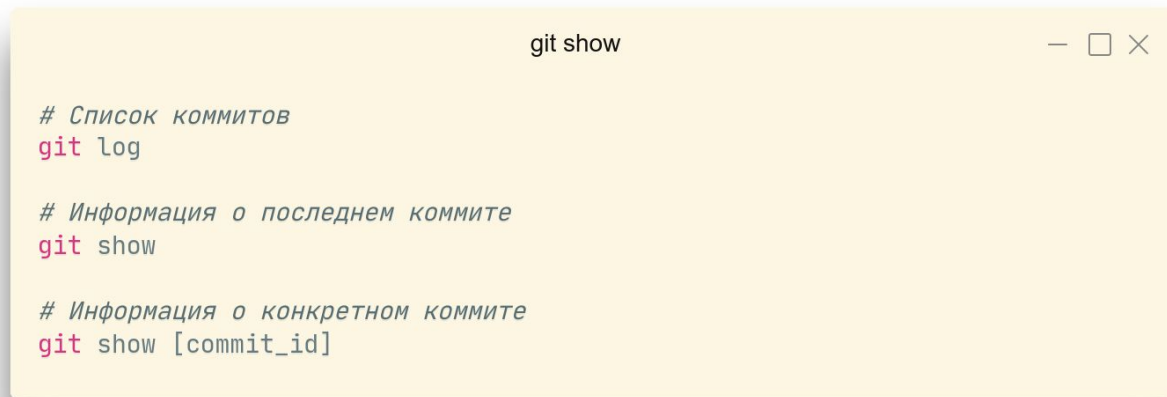
```
git add -A
```

Создаём commit

```
git commit -m "Сообщение о первом коммите"
```



Просмотр истории коммитов



```
git show

# Список коммитов
git log

# Информация о последнем коммите
git show

# Информация о конкретном коммите
git show [commit_id]
```

Возврат по истории коммитов

Checkout



Вернуть состояние на [commit_id]

```
git checkout [commit_id]
```

Загрузить состояние последнего коммита ветки master

```
git checkout master
```



Оформление сообщения о коммите

- Автор, email
- Описание причин и сути изменений
- Ссылка на задачу в трекере
- Дата

Полный цикл коммита

git commit full



```
# Редактируем файл  
echo "Test new git commit" >> file.txt  
  
# Смотрим, что изменилось  
git diff  
  
# Проиндексированные изменения  
git diff --cached  
  
# Фиксируем изменения  
git commit -am "String added!"  
  
# Смотрим историю  
git log
```



Работа с GitHub

Связь с удалённым репозиторием

```
GitHub init

# Новый репозиторий - подключить к GitHub
echo "# otus_test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin
git@github.com:Nickmob/otus_test.git
git push -u origin main
```

```
# Залить существующий репозиторий в GitHub
git remote add origin
git@github.com:Nickmob/otus_test.git
git branch -M main
git push -u origin main
```

Ключи для доступа к GitHub

```
git keys

# Генерируем ключи
ssh-keygen

# Копируем публичный ключ
cat ~/.ssh/id_rsa.pub

# Вставляем на https://github.com/settings/keys
```



Работа с ветками

git branch



```
# Создаём ветку
git branch feature

# Переходим в неё
git checkout feature

git add .
git commit -m 'Add new file1'

# Выливка изменений на Github
git push origin master (main)

# Смотрим удалённые репозитории
git remote -v
```



Получение кода из GitHub

master 1 branch 0 tags

Go to file Add file Code

vi-homutov and morf QUIC: fixed using of retired connection id (ticket #2289)

auto	UDP: fixed check UDP_SEGMENT	
conf	MIME: added image/avif type.	
contrib	Contrib: vim syntax, update core a	
docs	nginx-1.21.4-RELEASE	
misc	Updated OpenSSL used for win32 builds.	3 months ago
src	QUIC: fixed using of retired connection id (ticket #2289).	2 hours ago
README	README: HTTP/3 trailers are now supported.	3 months ago

Clone

HTTPS SSH GitHub CLI New

`https://github.com/VKCOM/nginx-quic.git`

Use Git or checkout with SVN using the web URL.

Download ZIP

git clone URL

Практика

Домашнее задание

1. Установить Git
2. Создать репозиторий с конфигами Nginx (из задания про веб)
3. Организовать хранение репозитория в GitHub
4. Настроить авторизацию по ключу с сервера на GitHub
- 5.* Автоматизировать получение конфига из GitHub



Сроки выполнения: указаны в личном кабинете



Что мы изучили?

Подведем итоги

1. Принципы контейнеризации
2. Отличия от виртуализации
3. Базовые команды по управлению контейнерами в Docker
4. Создание образов в Docker



Список материалов для изучения

1. <https://onlywei.github.io/explain-git-with-d3/>
2. <https://github.com/>
3. https://learngitbranching.js.org/?locale=ru_RU
4. <https://ohshitgit.com/>
5. <http://chm.org.ua/git-interview/>

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет



Рефлексия

Цели вебинара

Проверка достижения целей

1. Разобрали, что управление версиями.
2. Посмотрели основные сценарии использования git.
3. Научились работать с GitHub.



Вопросы для проверки

1. Какие типы систем управления версиями вы знаете?
2. К какому типу относится git?
3. Какая команда позволяет скачать сторонний репозиторий?

Рефлексия



Что было самым полезным на занятии?



Как будете применять на практике то, что узнали на вебинаре?

Следующий вебинар



Сети. Базовые знания



Ссылка на вебинар
будет в ЛК за 15 минут



Материалы
к занятию в ЛК —
можно изучать



Обязательный материал
обозначен красной
лентой



**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Лавлинский Николай

Технический директор “Метод Лаб”

<https://www.methodlab.ru/>

<https://www.youtube.com/c/NickLavlinsky>

<https://vk.com/nick.lavlinsky>

