Системы контроля версий

VCS — Version Control System

Зачем нужны системы контроля версий

- Как не потерять файлы с исходным кодом?
- Как защититься от случайных исправлений и удалений?
- Как отменить изменения, если они оказались некорректными?
- Как одновременно поддерживать рабочую версию и разработку новой?

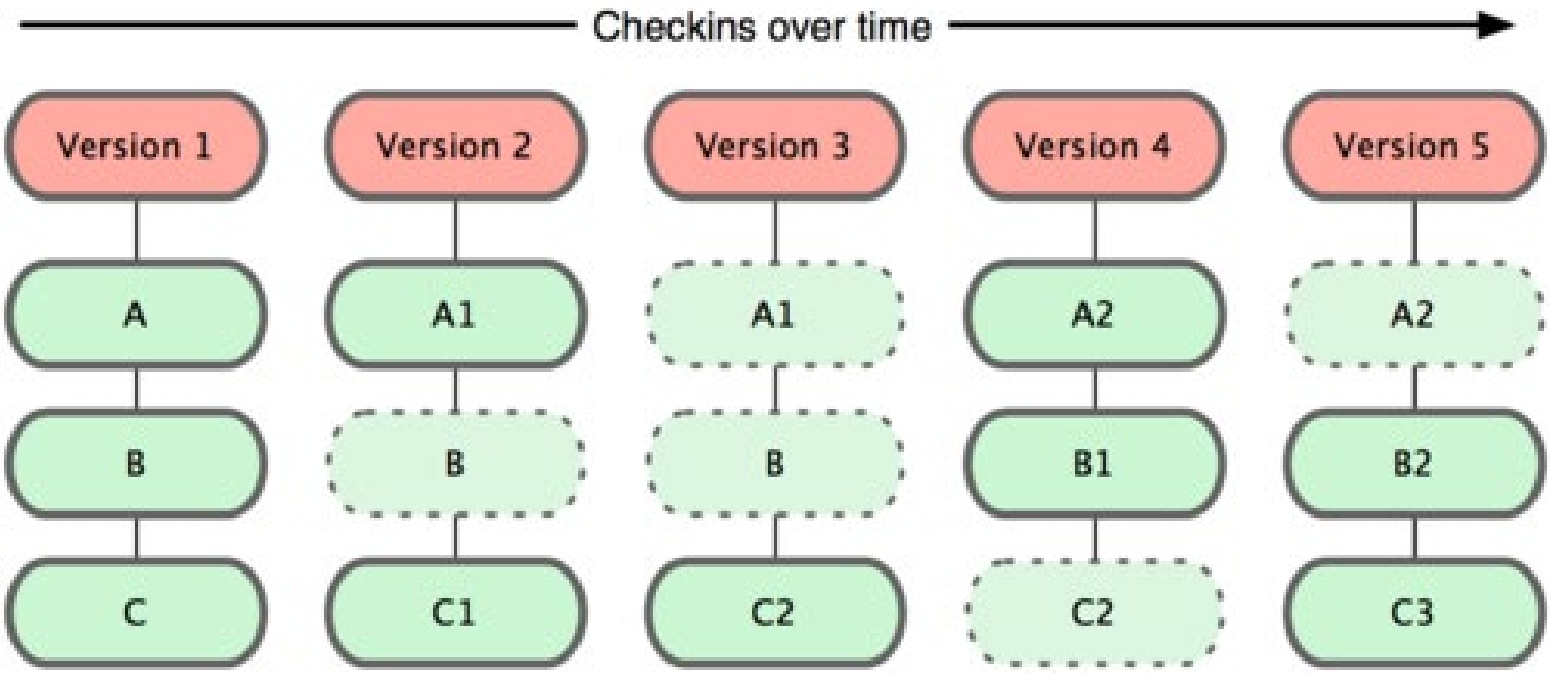
Возможности VCS:

- Возврат к любой версии кода из прошлого.
- Просмотр истории изменений.
- Совместная работа без боязни потерять данные или затереть чужую работу.

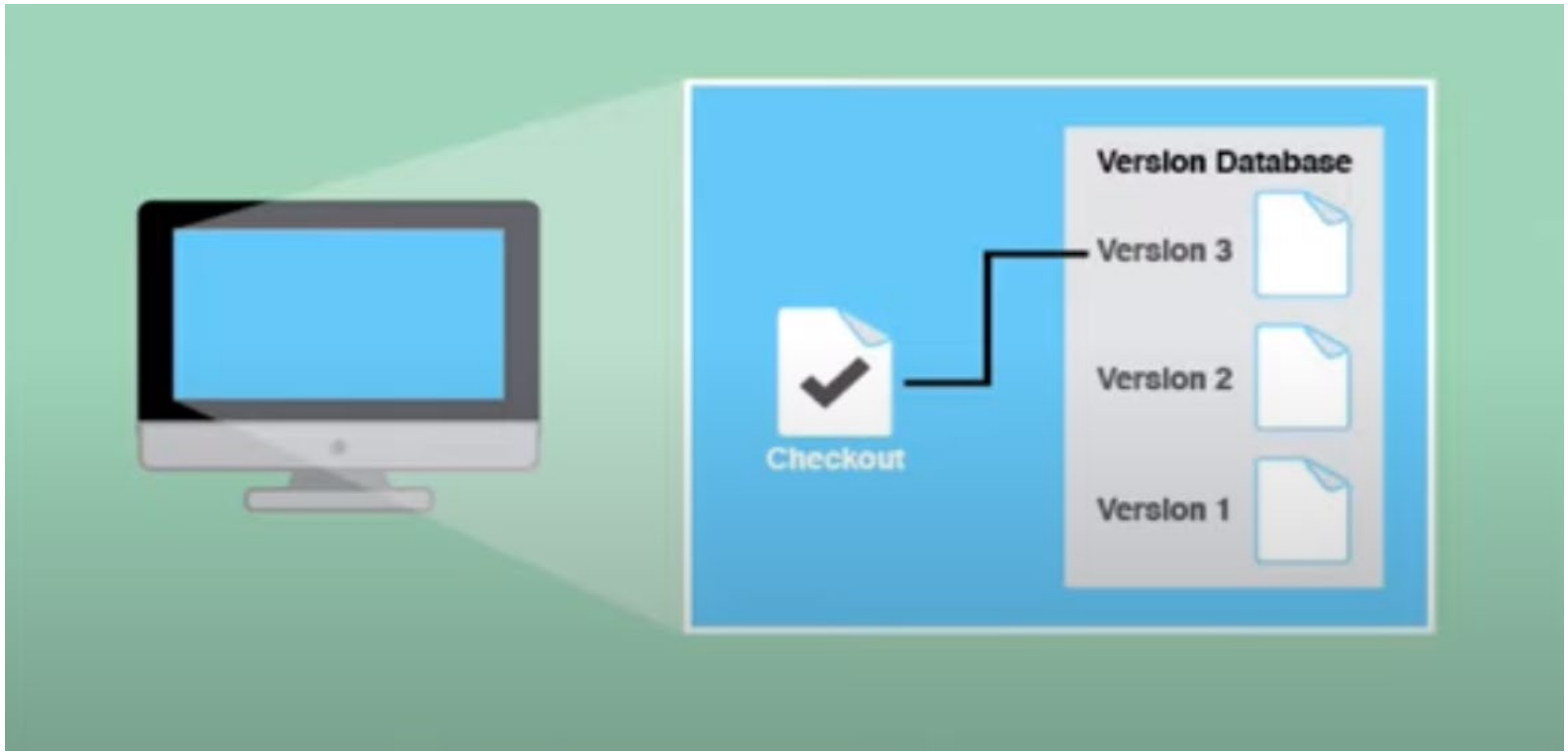
GIT

Git воспринимает свои данные скорее как набор снимков мини-файловой системы Каждый раз, когда вы фиксируете или сохраняете состояние своего проекта в Git, он в основном делает снимок того, как выглядят все ваши файлы в данный момент, и сохраняет ссылку на этот снимок.

Чтобы быть эффективным, если файлы не изменились, Git не сохраняет файл снова—просто ссылка на предыдущий идентичный файл, который он уже сохранил.

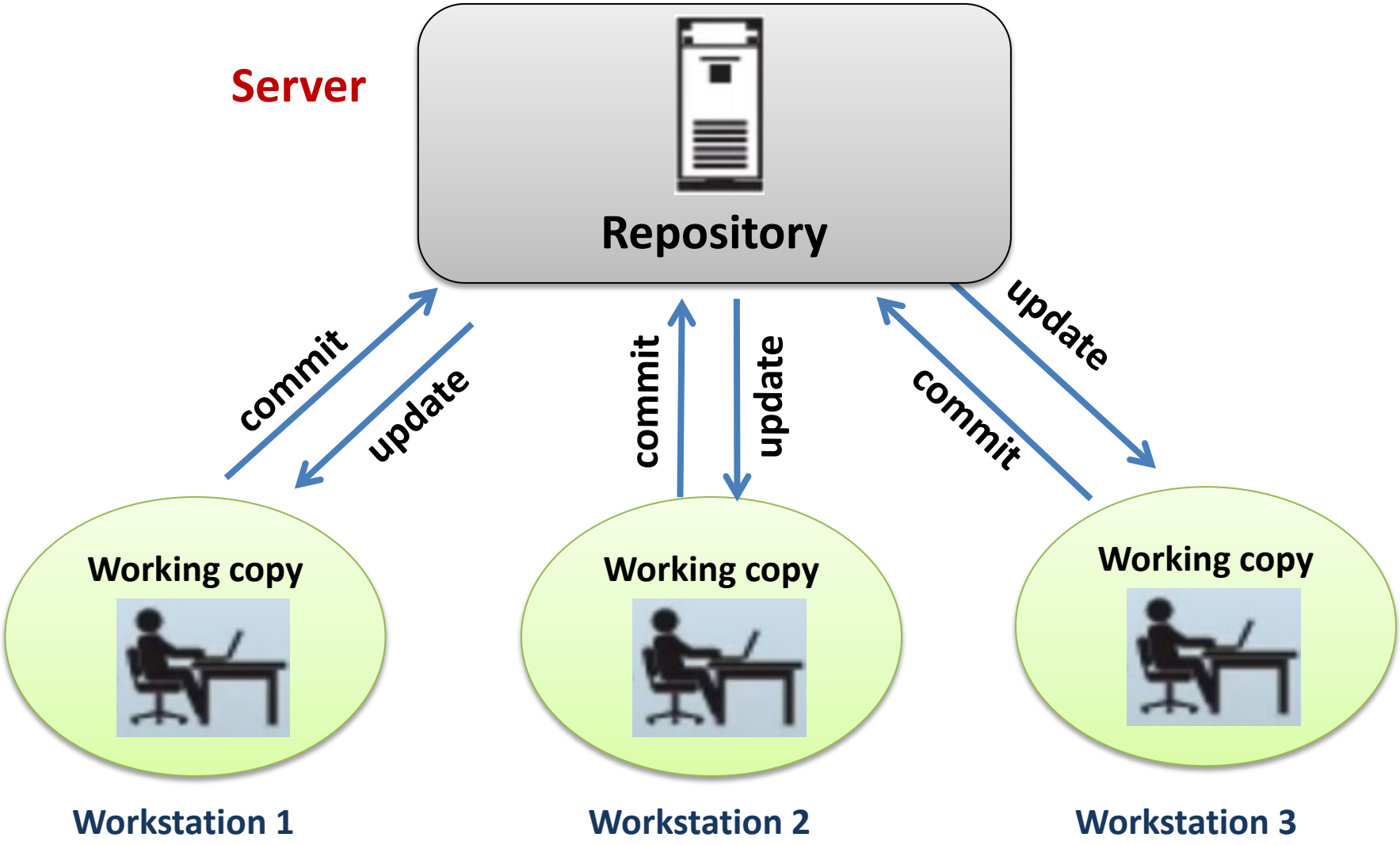


Локальные СКВ *(RCS, SCCS)*



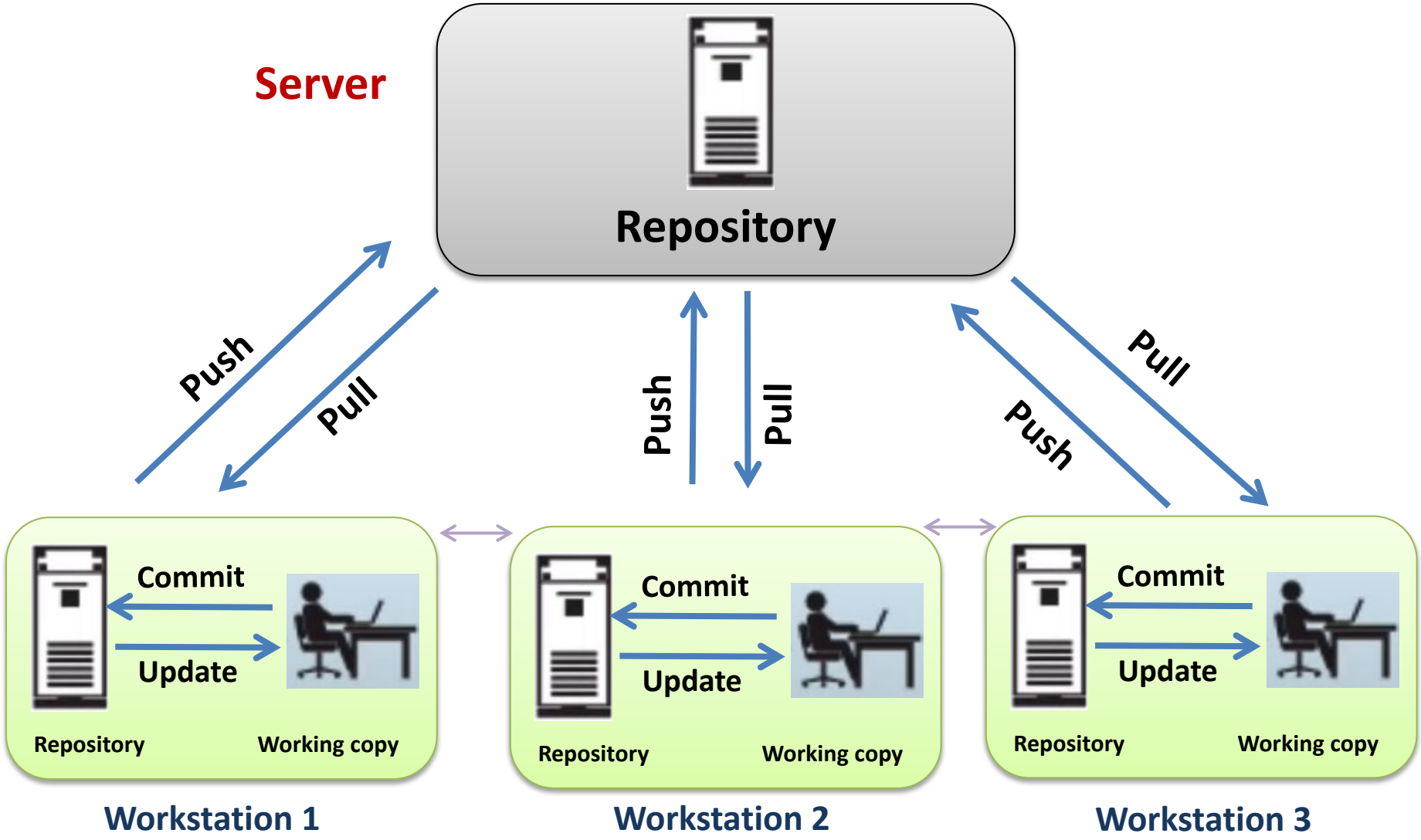
Централизованные СКВ (*CVS, SVN*)

Centralized version control system



Распределенные СКВ (*GIT, Mercurial*)

Distributed version control system



Основные понятия GIT

Ветка (branch)

- master - это, как правило, основная ветка проекта. Она появляется сразу после клонирования или инициализации репозитория. От него ответвляются другие
- Branch - соответствует состоянию родительского объекта на момент ветвления

Домашнее задание

- Создать аккаунт на www.github.com
- Создать public repository
- Установить git: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- Указать имя пользователя и email

  git config --global user.name "John Doe"

  git config --global user.email johndoe@example.com
- На компьютере через CMD или Terminal создать папку с любым именем
- В данной папке выполнить инструкции по созданию репозитария – **…or create a new repository on the command line**
  - Инициализировать локальный репозитарий
  - Создать файл в папке README.md
  - Добавить фал в репозитарий
  - Сделать commit (фиксация изменений в ветке)
  - Отправить изменения в удаленный репозитарий

Дополнительное задание:

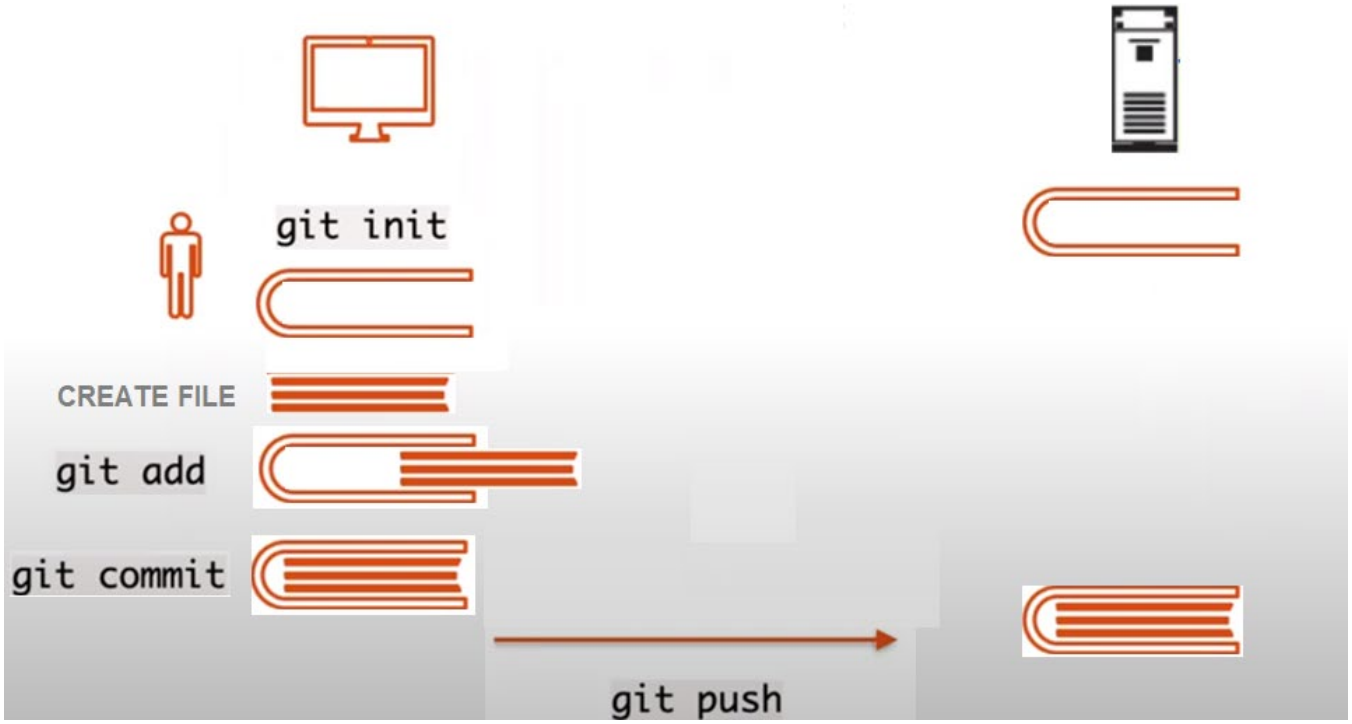
- Изменить файл README
- Добавить любой новый файл
- Обновить удаленный репозитарий и убедиться, что он обновился

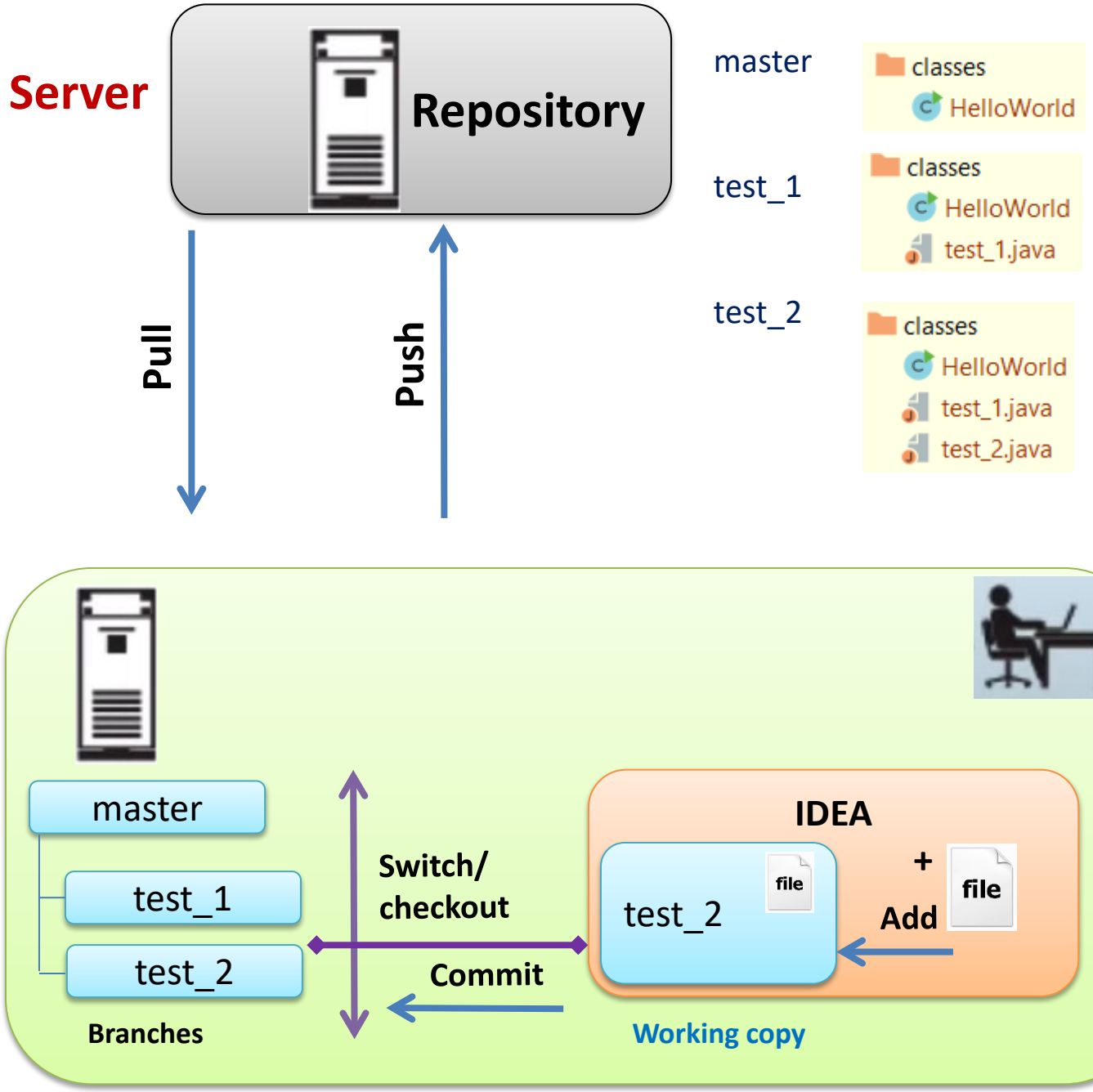
Книга по GIT: https://git-scm.com/book/en/v2

– …or create a new repository on the command line

```
echo "# 111" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/...
git push -u origin main
```

- Создать файл README.md
- Инициализировать локальный репозитарий с пустой веткой master
- Добавить фал в репозитарий в ветку master
- Сделать commit (фиксация изменений в ветке)
- Переименовать ветку main в master
- Связать локальный репозитарий с удаленным
- Отправить изменения в удаленный репозитарий




Server
Repository
Pull
Push
master
classes
HelloWorld
test_1
classes
HelloWorld
test_1.java
test_2
classes
HelloWorld
test_1.java
test_2.java
master
test_1
test_2
Branches
Switch/
checkout
Commit
IDEA
test_2
file
+
Add
file
Working copy