FH Salzburg

Grundlagen der Informatik

Labor/Übung 02: Versionsverwaltung

Technik
Gesundheit
Medien

Was ist Versionsverwaltung (VCS)?

- Zur Verwaltung von Änderungen
- Ausschließlich für nicht-binär Files! (→ Textfiles jeder Art, z.B. Quellcode, ...)
- Wichtig in Softwareentwicklung, wissenschaftlichen Arbeiten und Projektmanagement

Wozu wird Versionsverwaltung verwendet?

- Nachvollziehbarkeit: alle Änderungen und durchführender User über Zeit
- Wiederherstellbarkeit: beliebige, alte Versionen können wiederhergestellt werden
- Zusammenarbeit: gleichzeitige Bearbeitung durch mehrere Personen

Begriffsdefinitionen

- Repository: zentrale Ablage für Dateien und Historie
- Commit: gespeicherte Momentaufnahme von Dateien zu bestimmtem Zeitpunkt
- Merge: Zusammenführen von verschiedenen Änderungen
- Branches: parallele Entwicklungszweige für verschiedene Features/Versionen

Arten von VCS

- Lokale VCS
  - Speichert Versionen einer Datei auf lokalem System
  - Beispiel: einfaches Backup-Management mit Datei-Kopien

- Zentralisierte VCS (CVCS)
  - Einzelnes, zentrales Repository, auf das alle zugreifen
  - Beispiele: CVS, Suversion (SVN)
  - Vor-/Nachteile: einfach zu verwalten, Single Point of Failure

Arten von VCS

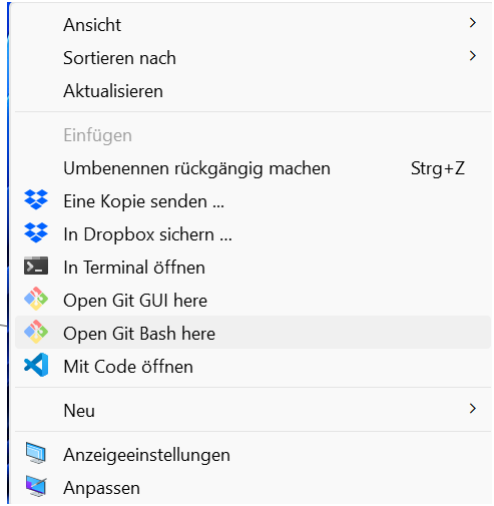
- Verteilte VCS (DVCS)
  - Jeder User hat Kopie von Repositories
  - Beispiele: git, Mercurial
  - Vorteile: keine Abhängigkeit von zentralen Server, schnelle Operationen, bessere Zusammenarbeit

git für Department IT

- https://its-git.fh-salzburg.ac.at/

- Zugriff mit fhsXXXXX@fh-salzburg.ac.at und Passwort

Versionsverwaltung – git



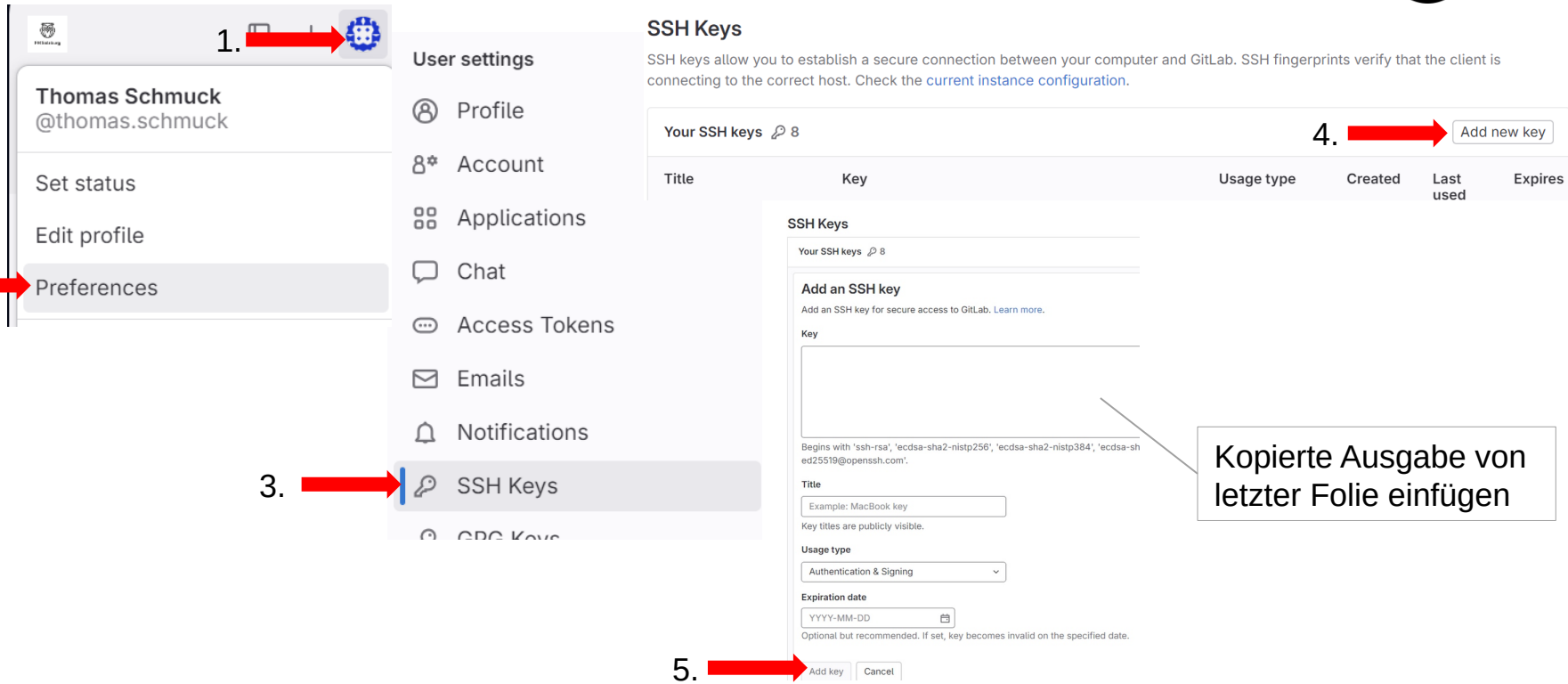
- https://git-scm.com/

Öffnen der git Bash

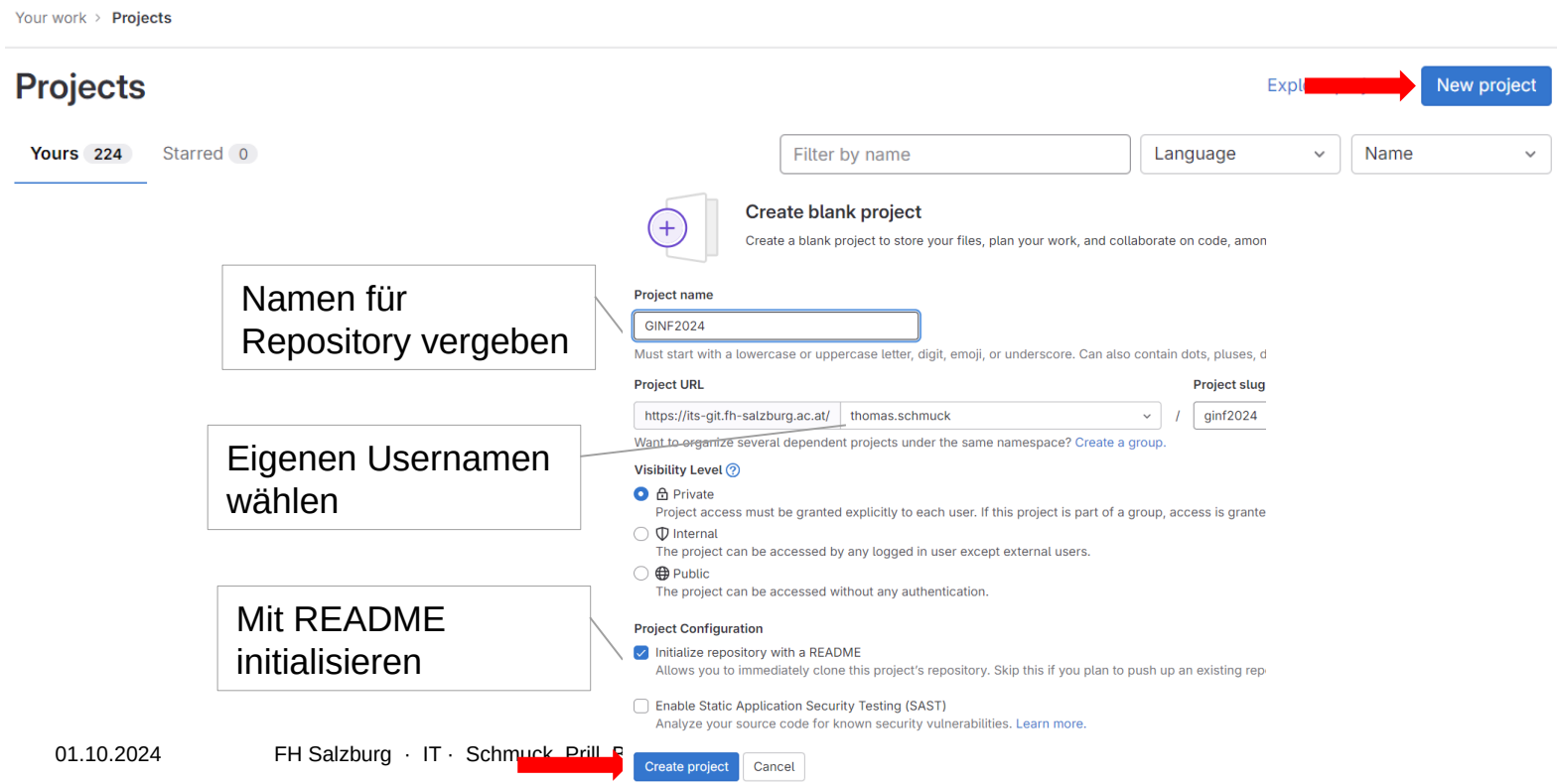
- `ssh-keygen` in Bash eingeben
- Bei Frage nach Pfad “Enter” drücken
- Bei Frage nach Passwort “enter” drücken
- `cat ~/.ssh/id_rsa.pub` eingeben und Ausgabe kopieren

```
thsch@Notebook MINGW64 ~/Desktop/git test
$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/thsch/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

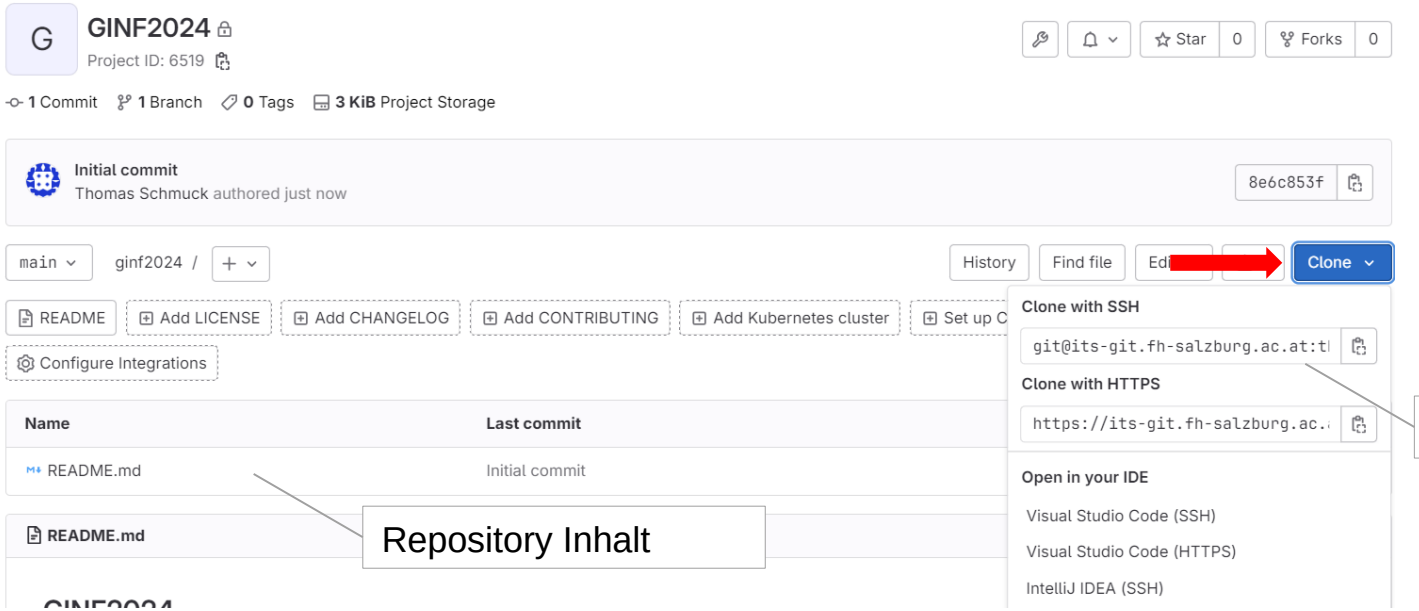
Versionsverwaltung – git



Versionsverwaltung – git



Versionsverwaltung – git



git Kommandos

- Konfiguration von Usernamen und E-Mail

```
git config --global user.name
“[Username]“
git config --global user.email “[E-Mail]“
```

```
thsch@Notebook MINGW64 ~/Desktop/git test/ginf2024 (main)
$ git config --global user.email "thomas.schmuck@fh-salzburg.ac.at"

thsch@Notebook MINGW64 ~/Desktop/git test/ginf2024 (main)
$ git config --global user.name "Harry"
```

git Kommandos

- Kopieren des remote Repositories auf Rechner

```
git clone [Pfad zu Repository]
```

```
thsch@Notebook MINGW64 ~/Desktop/git test
$ git clone git@its-git.fh-salzburg.ac.at:thomas.schmuck/ginf2024.git
Cloning into 'ginf2024'...
The authenticity of host 'its-git.fh-salzburg.ac.at (193.170.193.10)' can't be e
stablished.
RSA key fingerprint is SHA256:iBD/WlV7jswGtugHQ3DS8jvOAJn+wkkFOQdMEUHAtnY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'its-git.fh-salzburg.ac.at' (RSA) to the list of know
n hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

git Kommandos

- Zum Wechseln in das Repository

```
cd [Repository Name]
```

```
thsch@Notebook MINGW64 ~/Desktop/git test
$ cd ginf2024/

thsch@Notebook MINGW64 ~/Desktop/git test/ginf2024 (main)
$
```

git Kommandos

- Geänderte Dateien hinzufügen

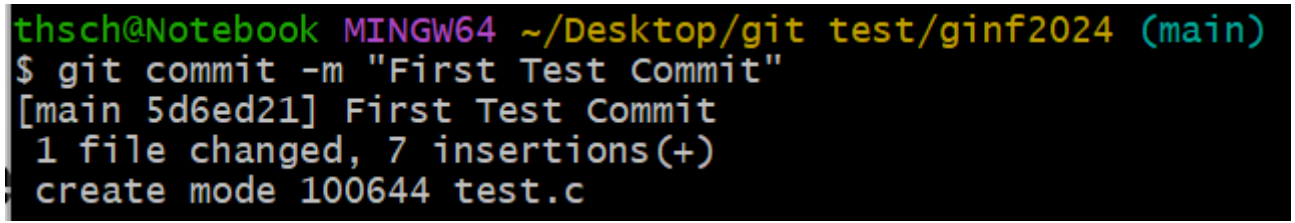
  `git add [Datei Namen]` (* für alle)

```
thsch@Notebook MINGW64 ~/Desktop/git test/ginf2024 (main)
$ git add *
```

git Kommandos

- Zum Hinzufügen einer zuvor hinzugefügten Änderung als Meilenstein

```
git commit -m “[Commit Message]“
```



git Kommandos

- Hochladen der Änderungen zu remote Repository

`git push`

```
thsch@Notebook MINGW64 ~/Desktop/git test/ginf2024 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 394 bytes | 394.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To its-git.fh-salzburg.ac.at:thomas.schmuck/ginf2024.git
   8e6c853..5d6ed21  main -> main
```

git Kommandos

- Herunterladen von Änderungen vom remote Repository

```
git pull
```

```
thsch@Notebook MINGW64 ~/Desktop/git test/ginf2024 (main)
$ git pull
Already up to date.
```

Best Practices

- Commit Messages: kurze und aussagekräftige Beschreibungen von Änderungen
- Regelmäßige Commits: kleine, häufige Commits für leichtere Nachvollziehbarkeit → eine abgeschlossene, selbstständige Änderung ist ein Commit
- Konfliktmanagement: frühzeitig lösen für einfacheren Integrationsprozess