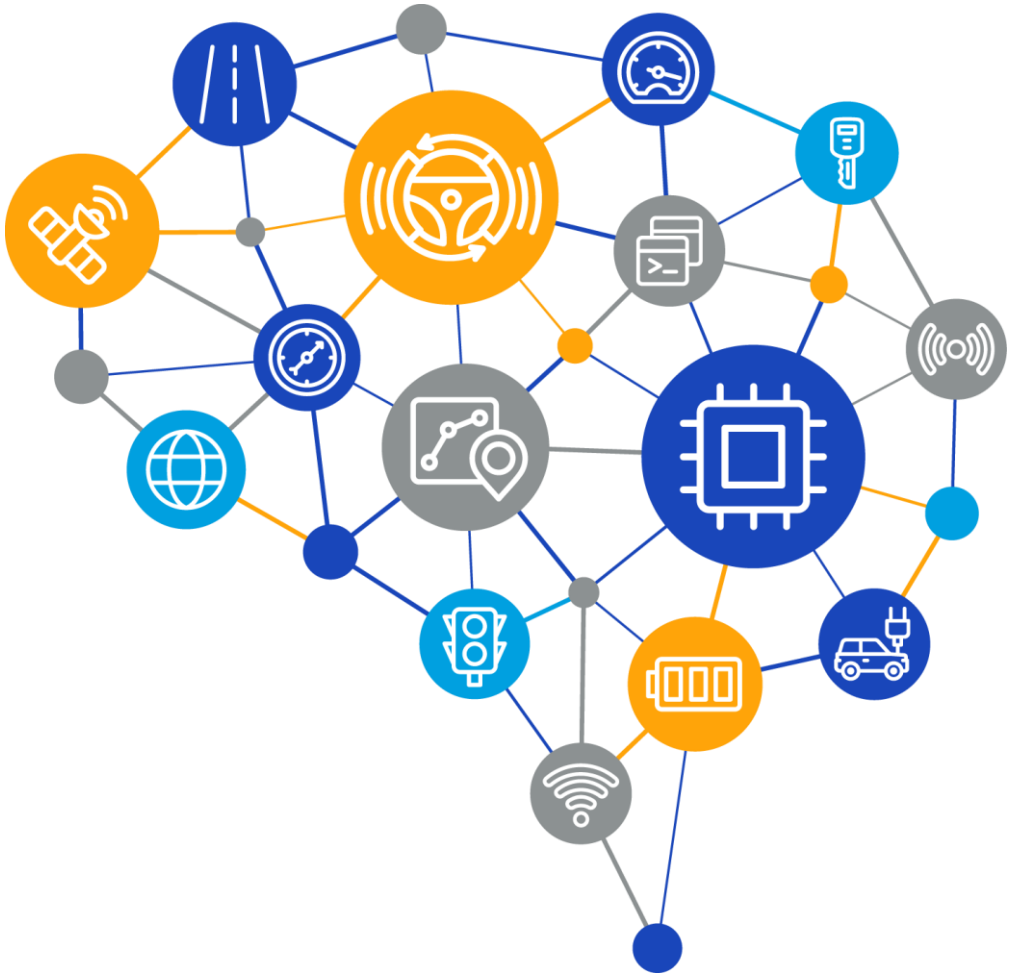
지역과 원격저장소에서 브랜치 생성과 연동 방법



강환수 교수

DMUAi 동양미래대학교 인공지능소프트웨어학과

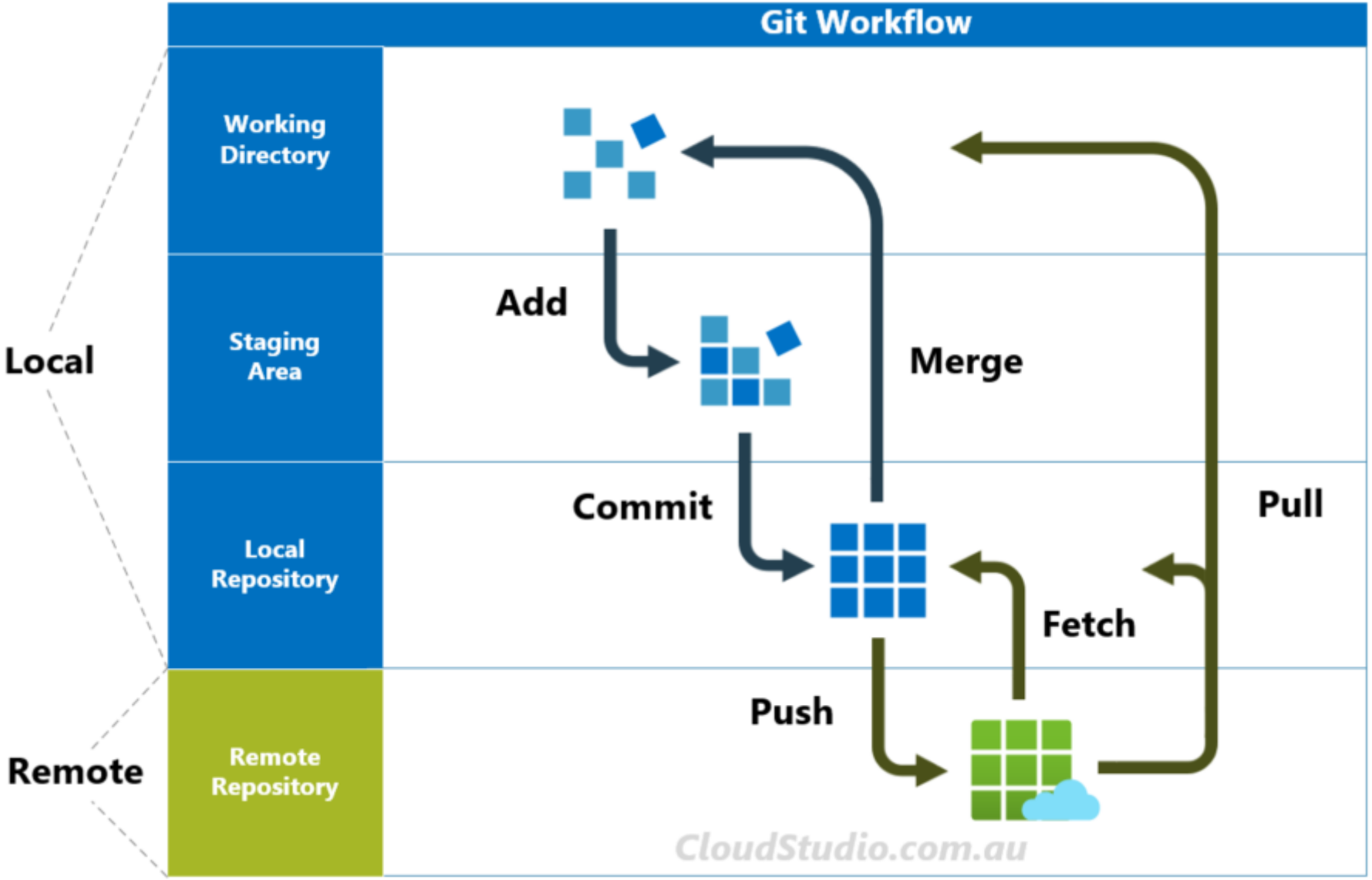
AI Experts Who Lead The Future

Goal

- **학습 개요**
  - 원격저장소와 지역저장소에서 여러 브랜치의 연동

- **학습 목표**
  - 원격저장소에서 만든 새로운 브랜치 지역저장소와 연동
  - 반대로 지역저장소에서 만든 새로운 브랜치 원격저장소와 연동

Push Pull


Git Workflow
Working Directory
Staging Area
Local Repository
Remote Repository
Local
Remote
Add
Commit
Push
Merge
Fetch
Pull
CloudStudio.com.au

AI Experts
Who Lead
The Future

01

Remote update와 pull

$ git remote update

- 모든 원격저장소의 브랜치 커밋 정보를 최신 상태로 갱신
  - 하지만, 지역저장소에 파일 수정을 반영하는 병합은 하지 않음
- $ git log --all
  - 원격저장소 브랜치 커밋 로그 변동 사항을 파악 가능

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline
81c9c88 (HEAD -> main, origin/main, origin/HEAD) Update README.md
74346f4 Initial commit

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git remote update
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 755 bytes | 83.00 KiB/s, done.
From https://github.com/ai7dnn/my-test
   81c9c88..b9ef8c4  main       -> origin/main

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
b9ef8c4 (origin/main, origin/HEAD) Update README.md
81c9c88 (HEAD -> main) Update README.md
74346f4 Initial commit
```

원격저장소 브랜치인 origin/main을 의미

원격저장소 브랜치의 HEAD를 의미

원격저장소 브랜치에서 커밋이 발생해 커밋이 하나 더 많아짐

$ git pull

- 현재 위치하고 있는 원격 브랜치를 업데이트하고 지역저장소에도 이 변동 사항을 병합

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
1efd90b (origin/main, origin/HEAD) Update remote update
b9ef8c4 Update README.md
81c9c88 (HEAD -> main) Update README.md
74346f4 Initial commit

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git pull origin main
From https://github.com/ai7dnn/my-test
 * branch            main       -> FETCH_HEAD
Updating 81c9c88..1efd90b
Fast-forward
 README.md | 9 ++++++++-
 1 file changed, 8 insertions(+), 1 deletion(-)

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ cat readme.md
# my-test
clone push fetch merge

### $ git clone [URL]
- 원격 저장소를 지역 저장소에 복제

### $ git pull
- 원격저장소의 내용을 지역저장소에 반영

### $ git remote update
- 모든 원격 브랜치를 업데이트하여 최신 상태로 갱신


PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
1efd90b (HEAD -> main, origin/main, origin/HEAD) Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit
```

원격이 2개 더 커밋이 발생한 상황

$ git pull 도 가능

서버에서 수정한 내용이 지역에도 반영되어 표시

원격 브랜치의 앞선 커밋으로의 이동도 가능

- $ git switch –d
- $ git checkout

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
f81485c (origin/main, origin/HEAD) switch branch
1efd90b (HEAD -> main, update) Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit
```

원격저장소 브랜치에서 커밋이 발생해 커밋이 하나 더 많은 상태

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git switch -d f814
HEAD is now at f81485c switch branch
```

원격저장소 브랜치의 앞선 커밋으로 이동

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test ((f81485c...))
$ git log --oneline
f81485c (HEAD, origin/main, origin/HEAD) switch branch
1efd90b (update, main) Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit
```

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test ((f81485c...))
$ cat readme.md
# my-test
clone push fetch merge

### $ git clone [URL]
- 원격 저장소를 지역 저장소에 복제

### $ git pull
- 원격저장소의 내용을 지역저장소에 반영

### $ git remote update
- 모든 원격 브랜치를 업데이트하여 최신 상태로 갱신

```
병합하지 않고 원격저장소의 내용을 최신 내용을 볼 수 있나요?
```
```

원격저장소 브랜치에서
수정된 내용을 먼저 확인할 수 있음

$ git push

- **지역저장소 수정 내용을 원격저장소에 반영**
  - 항상 pull을 선행이 되어야 함

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git commit -m add hello.py
[main 91c9200] add
 1 file changed, 1 insertion(+)
 create mode 100644 hello.py

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
91c9200 (HEAD -> main) add
f81485c (origin/main, origin/HEAD) switch branch
1efd90b Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 267 bytes | 267.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ai7dnn/my-test.git
   f81485c..91c9200  main -> main

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git push
Everything up-to-date
```

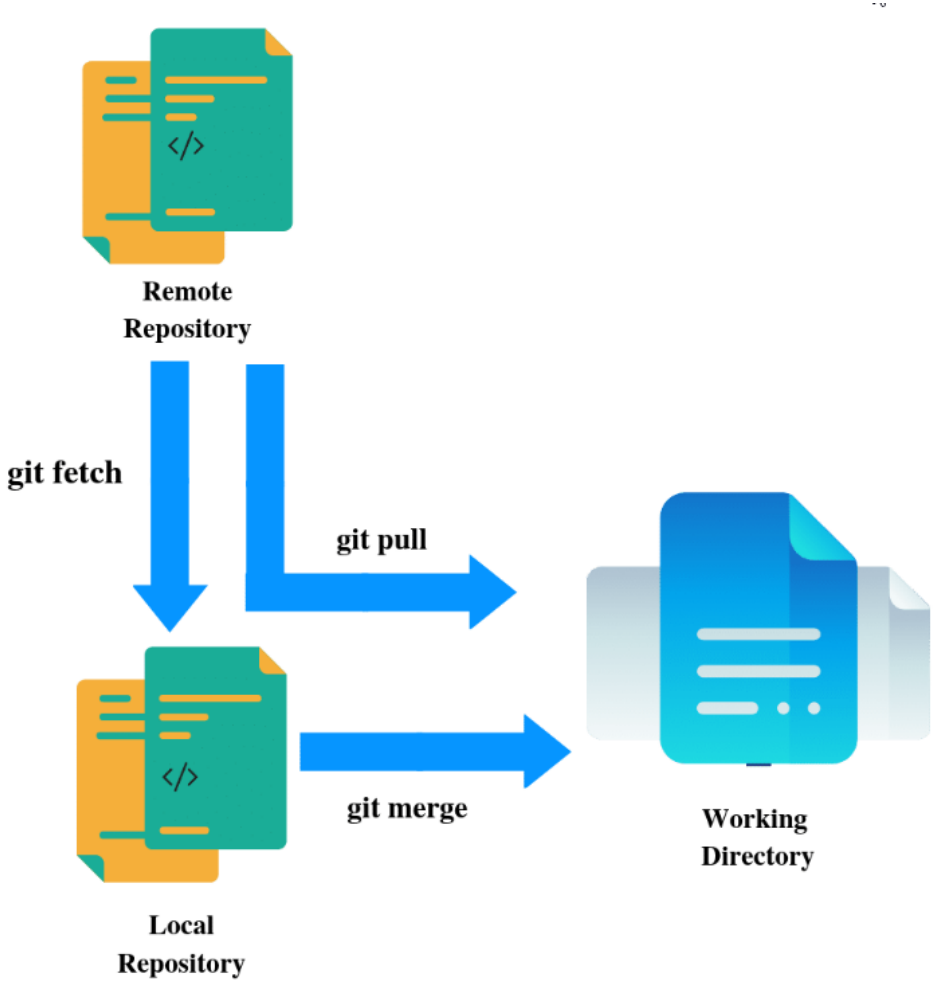
지역저장소에서 커밋 발생

붉은 색 원격저장소보다 지역저장소의 커밋이 하나 더 발생한 정보가 표시

지역과 원격이 도일해 더 이상 push 할 내용이 없는 경우 결과 표시

$ git fetch

- 현재 위치하고 있는 원격의 모든 브랜치의 커밋 정보를 수정
  - 하지만, 로컬에서 커밋의 파일 내용까지 병합하지는 않음
- 옵션 --all
  - 모든 원격 브랜치의 커밋 정보를 수정



$ git fecth origin main

- 원격별칭과 브랜치명까지 입력

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git fetch origin main
From https://github.com/ai7dnn/my-test
 * branch            main       -> FETCH_HEAD

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git fetch

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
cde07e7 (origin/main, origin/HEAD) Update README.md fetch
91c9200 (HEAD -> main) add
f81485c switch branch
1efd90b Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit
```

병합전 파일 차이

- **$ git diff [이전커밋ID] [이후커밋ID]**

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git diff main origin/main
diff --git a/README.md b/README.md
index a5092e6..db2acb4 100644
--- a/README.md
+++ b/README.md
@@ -13,3 +13,7 @@ clone push fetch merge
 ```
 병합하지 않고 원격저장소의 내용을 최신 내용을 볼 수 있나요?
 ```
+
+### $ git fetch
+- 현재 위치하고 있는 원격저장소 브랜치만 업데이트, 하지만, 로컬에서 변동 사항을 병합하지는 않음
+

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git diff 91c9 cde0
diff --git a/README.md b/README.md
index a5092e6..db2acb4 100644
--- a/README.md
+++ b/README.md
@@ -13,3 +13,7 @@ clone push fetch merge
 ```
 병합하지 않고 원격저장소의 내용을 최신 내용을 볼 수 있나요?
 ```
+
+### $ git fetch
+- 현재 위치하고 있는 원격저장소 브랜치만 업데이트, 하지만, 로컬에서 변동 사항을 병합하지는 않음
+
```

병합

- **$ git merge**

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git branch
* main

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git branch --all
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git merge
Updating 91c9200..cde07e7
Fast-forward
 README.md | 4 ++++
 1 file changed, 4 insertions(+)

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
cde07e7 (HEAD -> main, origin/main, origin/HEAD) Update README.md fetch
91c9200 add
f81485c switch branch
1efd90b Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit
```

원격이 하나라면 $ git remote update $ git fetch 동일

- 변화한 것이 없으면 결과는 아무 것도 표시되지 않음

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline
cde07e7 (HEAD -> main, origin/main, origin/HEAD)
Update README.md fetch
91c9200 add
f81485c switch branch
1efd90b Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git fetch origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-
reused 0
Unpacking objects: 100% (3/3), 786 bytes | 78.00
KiB/s, done.
From https://github.com/ai7dnn/my-test
 * branch            main       -> FETCH_HEAD
   cde07e7..c1864f6  main       -> origin/main

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
c1864f6 (origin/main, origin/HEAD) Update README.md
merge
cde07e7 (HEAD -> main) Update README.md fetch
91c9200 add
f81485c switch branch
1efd90b Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git remote update

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
c1864f6 (origin/main, origin/HEAD) Update README.md
merge
cde07e7 (HEAD -> main) Update README.md fetch
91c9200 add
f81485c switch branch
1efd90b Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit
```

fetch후 pull을 해도 오류 발생 없이 잘 수행

- 그러나 fetch한 후에는 merge로 병합

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
c1864f6 (origin/main, origin/HEAD) Update README.md merge
cde07e7 (HEAD -> main) Update README.md fetch
91c9200 add
f81485c switch branch
1efd90b Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused
0
Unpacking objects: 100% (3/3), 732 bytes | 73.00 KiB/s,
done.
From https://github.com/ai7dnn/my-test
   c1864f6..c33df53  main       -> origin/main

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
c33df53 (origin/main, origin/HEAD) Update README.md fetch
+ pull
c1864f6 Update README.md merge
cde07e7 (HEAD -> main) Update README.md fetch
91c9200 add
f81485c switch branch
1efd90b Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git pull
Updating cde07e7..c33df53
Fast-forward
 README.md | 7 +++++++
 1 file changed, 7 insertions(+)

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/my-test (main)
$ git log --oneline --all
c33df53 (HEAD -> main, origin/main, origin/HEAD) Update
README.md fetch + pull
c1864f6 Update README.md merge
cde07e7 Update README.md fetch
91c9200 add
f81485c switch branch
1efd90b Update remote update
b9ef8c4 Update README.md
81c9c88 Update README.md
74346f4 Initial commit
```

AI Experts
Who Lead
The Future

02

지역저장소에서 생성한 브랜치를 원격저장소에 반영

명령 git push –u [원격별칭] [새브랜치명]

- 지역저장소에서 새로 만든 저장소 hotfix를 원격별칭 origin에 올리려면
  - $ git push –u[--set-upstream] origin hotfix

- 다음 설정을 지정하면
  - $ git config --global push.autoSetupRemote true
  - $ git push
    - 새로 만든 hotfix 에서 위 명령으로도 가능

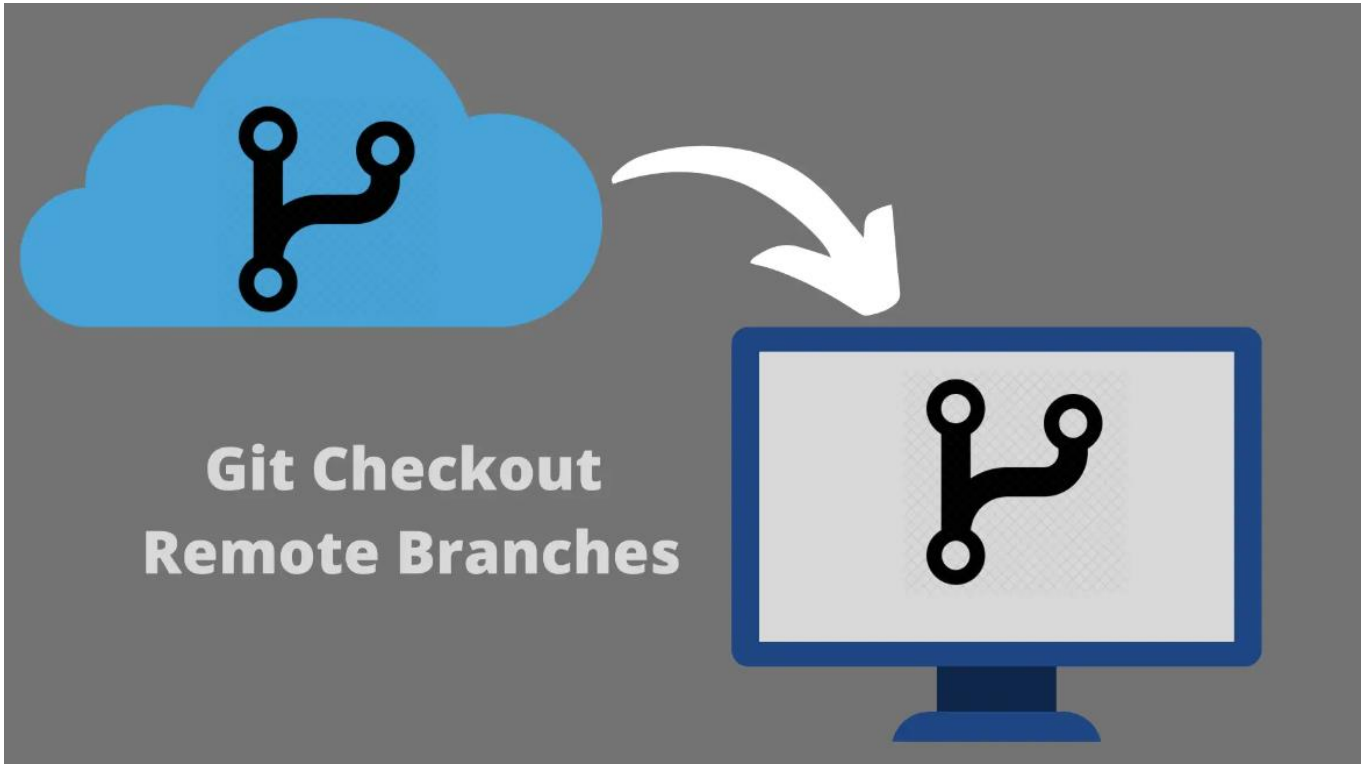
AI Experts
Who Lead
The Future

03

Git Checkout Remote Branch (fetch 후 지역 브랜치 생성)

원격 브랜치를 지역 브랜치로 생성

- **명령 branch, checkout, switch 활용**
  - 옵션 --track 간단히 -t



먼저 fetch가 필요

- 명령 fetch는 다음처럼 브랜치 이름을 지정하거나 모두 가져오거나
  - $ git fetch
    - 현재 위치의 브랜치가 속한 원격별칭의 모든 브랜치 정보를 수정
  - $ git fetch [원격별칭] [브랜치명]
    - 특정 브랜치(원격에서 만든 hotfix 브랜치)의 커밋 정보 가져오기
      - $ git fetch origin hotfix

- 원격 별칭이 여러 개 있다면 옵션 --all로 모두 가능
  - $ git fetch --all
    - 다음으로도 가능, 다음은 같은 기능의 명령
  - $ git remote update
    - 등록된 모든 원격 저장소의 정보를 최신 상태로 수정

명령 git push –u [원격별칭] [새브랜치명]

- 위 문장 이후 원격저장소 remote branch를 원격 추적하는 지역 저장소 생성 방법
  - $ git branch new_branch –t[--track] origin/new_branch
    - $ git branch hotfix –t[--track] origin/hotfix
      - 이동은 안함
  - $ git checkout –b new_branch –t[--track] origin/new_branch
    - $ git checkout –b hotfix –t[--track] origin/hotfix
    - $ git switch -c hotfix –t[--track] origin/hotfix
  - $ git checkout –b new_branch origin/new_branch
    - $ git checkout –b hotfix origin/hotfix
    - $ git switch -c hotfix origin/hotfix : 검사해야함
  - $ git checkout –t[--track] origin/hotfix
    - $ git checkout –t[--track] origin/hotfix
    - $ git switch -t[--track] origin/hotfix
      - 원격과 동일한 이름으로 생성
  - $ git checkout new_branch
    - $ git checkout hotfix
      - 원격과 동일한 이름으로 생성
      - 원격이 origin/hotfix이면 hotfix로 입력

깃허브에서 여러 개의 브랜치를 만들어 지역에 복사

- 깃허브에서 main 하부에 4개 이상의 브랜치를 만들어 지역에 복사
  - 원격저장소 브랜치를 다음과 같이 생성
  - 복제 clone에 의해 main만 모두 내려 받으며 나머지 브랜치는 커밋 정보만 내려 받음

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]
$ git clone https://github.com/ai7dnn/my-r.git
Cloning into 'my-r'…
Remote: Enumerating objects: 12, done.
Remote: Counting objects: 100% (12/12), done.
Remote: Compressing objects: 100% (9/9), done.
remote: Total 12 (delta 0), reused 0 (delta 0),
pack-reused 0
Receiving objects: 100% (12/12), done.

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]
$ cd my-r

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r
(main)
$ ls
README.md
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (main)
$ git lag
* f011b82 (origin/h4) Create h4.md
| * 695d962 (origin/h3) Create hy3.md
|/
| * 6f2fb19 (origin/h2) Create h2.md
|/
| * 685a054 (origin/h1) Create h1.md
|/
* 78e39fd (HEAD -> main, origin/main, origin/HEAD) Initial commit

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (main)
$ git branch -a
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/h1
  remotes/origin/h2
  remotes/origin/h3
  remotes/origin/h4
  remotes/origin/main
```

하나의 원격 브랜치를 복제하고 추적(tracking) 연계

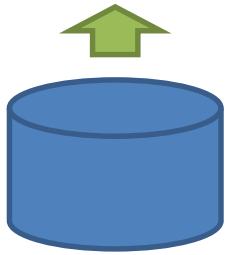
- **명령 1**
  - **$ git branch –t[--track] new_branch origin/new_branch**
    - **$ git branch –t[--track] hotfix origin/hotfix**
    - **다만, 브랜치 이동은 없음**

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (main)
$ git branch -t h1 origin/h1
branch 'h1' set up to track 'origin/h1'.

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (main)
$ git branch -a
  h1
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/h1
  remotes/origin/h2
  remotes/origin/h3
  remotes/origin/h4
  remotes/origin/main
```

Tracking branch는 로컬 브랜치 중에서 리모트 브랜치를 Tracking(추적)하는 브랜치

Remote Tracking branch는 origin/main과 같은 원격저장소의 브랜치를 참조하는 브랜치이며, 원격 브랜치의 복사본 같은 개념



Remote branch는 원격저장소의 브랜치로, github, gitlab같은 다른 원격 저장소에 존재하는 브랜치

- **$ git checkout –b new_branch origin/new_branch**
  - **$ git checkout –b hotfix origin/hotfix**
    - **다음으로도 가능, -t 추가**
  - **$ git checkout –t –b hotfix origin/hotfix**

- **명령 branch 옵션 –vv**
  - **각 로컬 브랜치들이 어떤 원격 브랜치를 추적(track) 하고 있는지를 확인**

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (main)
$ git checkout -b h2 origin/h2
Switched to a new branch 'h2'
branch 'h2' set up to track 'origin/h2'.

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (h2)
$ git branch -a -vv
  h1                    685a054 [origin/h1] Create h1.md
* h2                    6f2fb19 [origin/h2] Create h2.md
  main                  78e39fd [origin/main] Initial commit
  remotes/origin/HEAD   -> origin/main
  remotes/origin/h1     685a054 Create h1.md
  remotes/origin/h2     6f2fb19 Create h2.md
  remotes/origin/h3     695d962 Create hy3.md
  remotes/origin/h4     f011b82 Create h4.md
  remotes/origin/main   78e39fd Initial commit
```

- **$ git checkout –t[--track] origin/hotfix**
  - **$ git checkout –t[--track] origin/hotfix**

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (h2)
$ git checkout -t origin/h3
Switched to a new branch 'h3'
branch 'h3' set up to track 'origin/h3'.

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (h3)
$ git branch -a -vv
  h1                    685a054 [origin/h1] Create h1.md
  h2                    6f2fb19 [origin/h2] Create h2.md
* h3                    695d962 [origin/h3] Create hy3.md
  main                  78e39fd [origin/main] Initial commit
  remotes/origin/HEAD -> origin/main
  remotes/origin/h1     685a054 Create h1.md
  remotes/origin/h2     6f2fb19 Create h2.md
  remotes/origin/h3     695d962 Create hy3.md
  remotes/origin/h4     f011b82 Create h4.md
  remotes/origin/main   78e39fd Initial commit
```

- **$ git checkout new_branch**
  - **$ git checkout hotfix**

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (h3)
$ git checkout h4
Switched to a new branch 'h4'
branch 'h4' set up to track 'origin/h4'.

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/my-r (h4)
$ git branch -a -vv
  h1                  685a054 [origin/h1] Create h1.md
  h2                  6f2fb19 [origin/h2] Create h2.md
  h3                  695d962 [origin/h3] Create hy3.md
* h4                  f011b82 [origin/h4] Create h4.md
  main                78e39fd [origin/main] Initial commit
  remotes/origin/HEAD -> origin/main
  remotes/origin/h1   685a054 Create h1.md
  remotes/origin/h2   6f2fb19 Create h2.md
  remotes/origin/h3   695d962 Create hy3.md
  remotes/origin/h4   f011b82 Create h4.md
  remotes/origin/main 78e39fd Initial commit
```

원격 브랜치를 추적(track)하도록 upstream 정보를 설정

- 아래와 같이 입력
  - $ git branch --set-upstream-to=<remote>/<branch> <branch>
  - $ git branch -u <remote>/<branch> <branch>