

Workflow Git V2

```
$ git checkout master
$ git pull --rebase upstream master
$ git push origin master
```

} Synchronise upstream/master et origin/master
Référencé par \$synchro par la suite

} Je commence un nouveau dev

```
$ git checkout -b num-branchName
...
dev
...
$ git add [-A] .
$ git commit -m « #num - explication »
```

} Très souvent

```
...
$ synchro
$ git checkout num-branchName
$ git rebase -i master
$ git rebase master
```

} Au moins une fois par jour

```
...
$ git add [-A] .
$ git commit -m « #num - explication »
$ git push origin num-branchName
```

} Tous les soirs avant de partir

```
...
$ synchro
$ git checkout num-branchName
$ git rebase -i master
$ git rebase master
$ git push origin num-branchName
```

} J'ai fini mon dev :
- il est beau
- il a des tests
- il fait pas péter le build

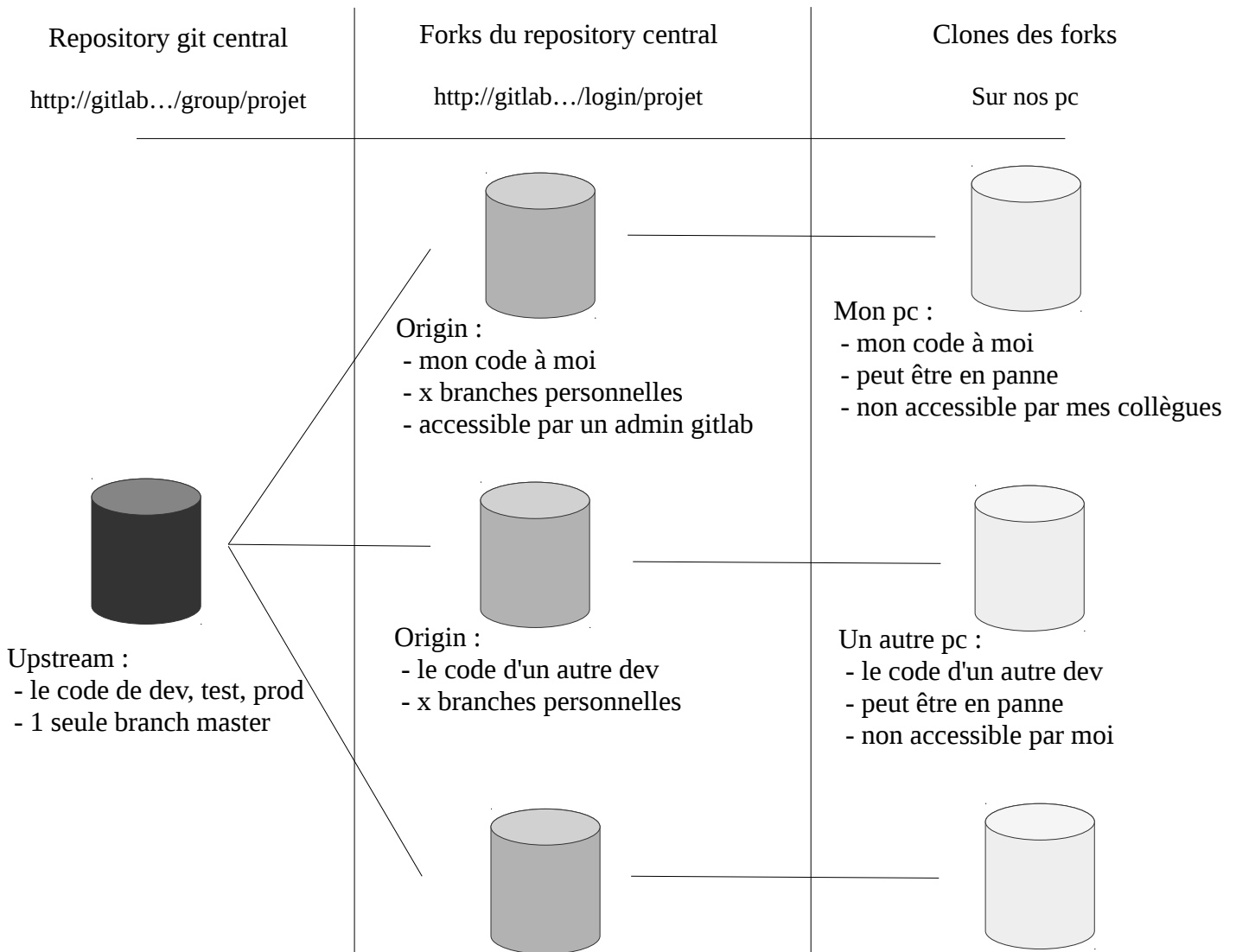
```
> aller sur http://gitlab.lan.itkweb.fr
> aller dans le projet « racine »
> onglet « Merge Requests »
> bouton « New Merge Request »
> dans « Source branch » sélectionner votrelogin/nomprojet puis la num-branchName
> dans « Target branch » sélectionner nomprojet/nomprojet puis la branch master
```

} Je propose mon dev aux autres devs, ils doivent faire la review et me donner des **conseils positifs** pour l'améliorer !

```
...
review du code et correction demandée
...
$ synchro
$ git checkout num-branchName
...
correction
...
$ git push origin num-branchName
...
review du code et acceptation, donc merge de num-branchName dans upstream/master
...
$ synchro
$ git branch -d num-branchName
$ git push origin :num-branchName
```

} C'est bon tout est fini !
Champagne & nettoyage !

Workflow Git V2



Origin → pc :
- git checkout branch
- git pull --rebase origin branch
Pc → Origin :
- git checkout branch
- git push origin branch

Origin → Upstream :
- « Merge Request » grâce à gitlab
Upstream → Origin :
- git checkout master
- git fetch upstream
- git rebase upstream/master
- git push origin master