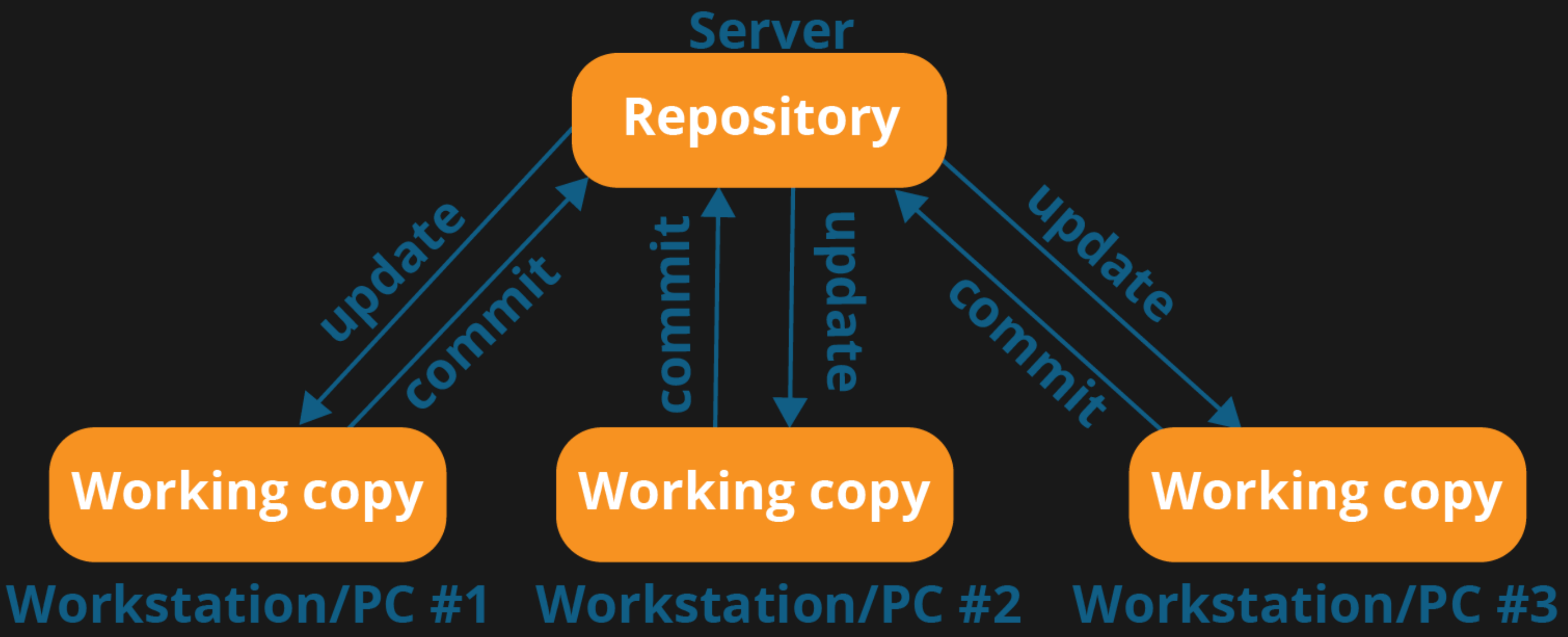
git

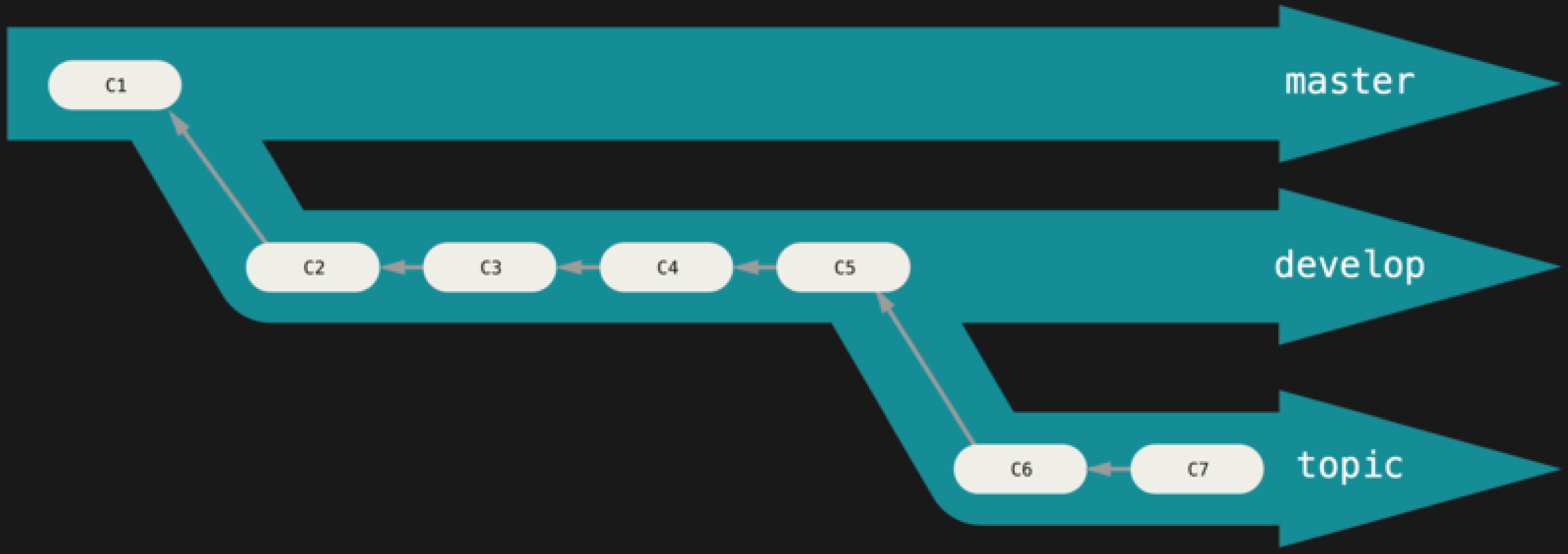
Free and open source distributed version control system

VERSION CONTROL

Centralized version control system



BRANCHING



EXERCISE 1 - INTRODUCTION

- Install git form here
- Create a repository

```
cd /your_project_folder
git init
```

- Check status

```
git status
```

- Begin tracking file

```
echo "hola" > filename.txt
git add filename
```

- Track all files

```
git add .
```

- Commit changes

```
git commit -m "message"
```

- Tag your commit

```
git tag -a v0.1 -m "version 0.1"
```

- Change between tags

```
# change the file and commit
# then go back to previous state

git checkout v0.1
```

- Delete a tag

```
git tag -d v0.1
```

EXERCISE 2 - BRANCHING

- List branches

```
git branch
```

- Create new branch

```
git checkout -b develop
```

- Change between branches

```
git checkout master
git checkout develop
```

- Merge a branch

```
# edit the file, commit changes
# go back to master

git merge develop
```

- Delete a branch

```
git branch -d develop
```

You can specify folder or files to NOT be tracked (usually big files or sensitive data) in the .gitignore file

```
echo "PASSWORD=myPassword" > .env
echo ".env" > .gitignore

# git now does not track .env files

git status
```

GitLab

EXERCISE 3 - GITLAB

- Create a Gitlab account
- Create a new repository on Gitlab
- Synchronize your local repository with Gitlab

```
git remote add origin https://gitlab.com/username/reponame.git
```

- Upload the changes

```
git push -u origin master
```

EXERCISE 4 - GITLAB

- Create a new repository on Gitlab
- Clone the repository to your machine

```
git clone https://gitlab.com/username/reponame.git
```

- Add some content
- Upload changes to remote repository

```
git push

# if you have tags
git push origin --tags
```

START COLLABORATING !

EXERCISE 5 - TEAM PROJECT

- Manager creates a new repository on Gitlab
- Manager adds collaborators
- Collaborators clone the repository
- Manager assign tasks to collaborators ... Issues
- Each collaborator works in its own branch
- Try to push changes ... Merge request
- Automate tasks with ... CI/CD (testing, deployment, generate docs, ...)
- And many more features like Project board , Labels, Wiki, Container registry ...

git