Évolution de l'outil informatique

calcul parallèle, données massives & développement collaboratif

Ecole de GIF, septembre 2019
David Chamont, LAL

Plan de bataille

- Grandes tendances
  - de 2000 à 2020, ce qui a changé.
- Projecteur sur ...
  - Calcul parallèle : programmation fonctionnelle, vectorisation SIMD, calcul déporté sur GPU.
  - Données massives : calcul distribué avec SPARK.
  - Développement collaboratif : Docker et Gitlab.
- Pour aller plus loin
  - outils et projets en cours à l'IN2P3.
  - ressources de formation.

Grandes tendances

2000-2020, ce qui a changé

Ecole de GIF, septembre 2019
David Chamont, LAL

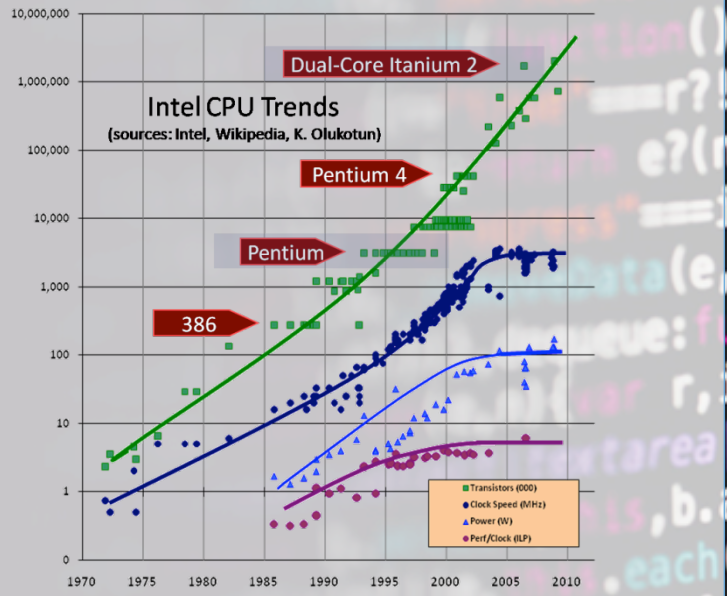
2000

- Acquisition, reconstruction et analyse
- Programmation orientée objet, en C++
- Un programme de reconstruction traite sequentiellement un lot d'évènements.
- L'ensemble des évènements est découpé en lots qui sont distribués sur une grille de calcul
- Gestion de version avec CVS

La fin du séquentiel candide

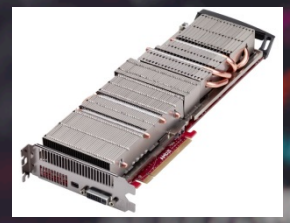
- Matériel : multiplication des coeurs
  - Stabilisation de la fréquence d'horloge.
  - Multiplication des cœurs, au sein du supercalculateur comme du téléphone portable.
  - La taille de la mémoire vive ne suit pas.
- Logiciel : parallélisme obligé
  - Fin des gains de performances gratuits.
  - Retour en grâce de la programmation concurrente.

« The free lunch is over »
Herb Sutter, 2005.



Un parallélisme multiforme

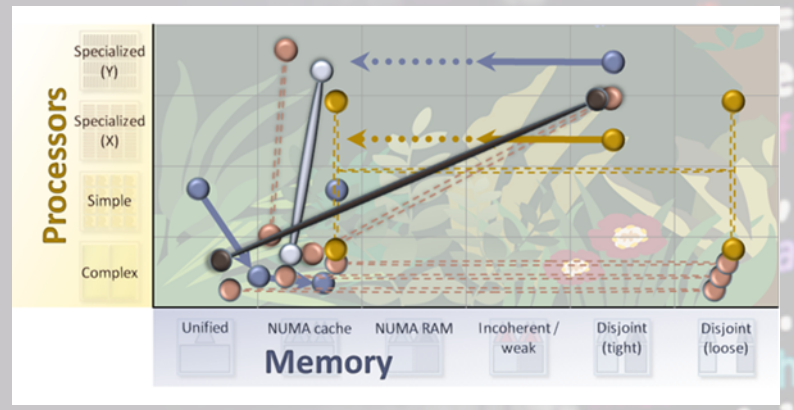
- Hors du serveur
  - Connection des serveurs d'un site par une réseau haut-débit : le supercalculateur.
  - Distribution des calculs sur de multiples sites via internet : la grille de calcul.
- Dans le serveur
  - Ajout de coeurs "lourds" programmés à travers des threads.
  - Dans ces coeurs, allongement des unités de traitement vectoriel.
  - Adjonction de coprocesseurs, regroupant des milliers de coeurs "légers".



Matériel hétérogène & programmation hybride

- Matériel : toujours plus complexe
  - Diversification, spécialisation et coopération des coeurs.
  - Tentatives de revenir vers une mémoire globale... mais non homogène.
- Logiciel : combiner les approches
  - Les accélérateurs ne peuvent pas tout faire. A combiner avec : instructions vectorielles des processeurs, multi-threading, aggrégation des nœuds d'un super-calculateur et/ou d'une grille.
  - Fin du dogme du double.

« Welcome to the jungle. »
Herb Sutter, 2013.



Les virtualisations s'empilent

- La grille imposait un système d'exploitation à l'utilisateur, limite que va lever le cloud.
- Les machines virtuelles, lourdes à mettre en oeuvre, vont laisser la place aux conteneurs.

Explosion du web & déluge de données

- Une nouvelle communauté se crée autour des Big Data et génère un nouvel éco-système logiciel, à commencer par Hadoop.
- Une conjonction entre quantité de données et progrès sur les GPGPUs amène le retour de l'apprentissage automatique.
- Les interfaces utilisateurs de déplacent sur le Web, voire sur les smartphones.
- Le phénomène des réseaux sociaux s'étend aux développeurs avec GitHub.

A la reconquête de la reproductibilité numérique

- La diversification des matériels et des précisions numériques, la parallélisation multiforme, les algorithmes monte-carlo, la masse des calculs et des données mettent à mal la reproductibilité des calculs, et donc de la science.
- Une résistance s'organise pour
  - Reprendre le contrôle des erreurs numériques
  - Archiver et DOIser les données et les logiciels
  - Tracer et automatiser toutes les étapes de calcul.

2020

- Acquisition ⇔ reconstruction ⇔ analyse
- Programmation fonctionnelle en Python
- Vectorisation & GPUs
- Déploiement de conteneurs Docker et Singularity
- Calcul distribué avec Spark
- Apprentissage automatique
- Développement collaboratif avec Gitlab
- *Diffusion de résultats par des notebooks*

2040

- Julia ?
- Informatique quantique ?
- Exascale
  - Un rapprochement de la mémoire et des processeurs ?
  - Des algorithmes sans communication ?
  - De la tolérance aux pannes et aux données incomplètes ?

Programmation fonctionnelle

pas seulement en Haskell

Ecole de GIF, septembre 2019
David Chamont, LAL

Une vision de la PF

- Se rapprocher de la logique mathématique :
  - une variable désigne toujours la même chose
  - avec les mêmes arguments, une fonction retourne toujours le même résultat ;
  - des méta-fonctions peuvent manipuler des fonctions ;
  - une algèbre des types permet de les recombiner.
- Compenser les inefficacités induites
  - évaluation paresseuse
  - structures de données immuables
  - dans les compilateurs C++ : l'optimisation de la valeur de retour (RVO), ...

Bénéfices de la PF

- En théorie
  - Prouvabilité
  - Code plus lisible
- En pratique :
  - Eviter les changements d'état involontaires
    Michael Feathers : OO makes code understandable by encapsulating moving parts. FP makes code understandable by minimizing moving parts.
  - Faciliter la parallélisation
    no shared states, no problem.

Un exemple C++

```
void count_lines_in_files( const vector<string> & files, vector<int> & nb_lines )  {
    vector<string>::iterator fileitr ;
    for ( fileitr = files.begin() ; fileitr != files.end() ; ++fileitr ) {
        int line_count = 0 ;
        char c = 0 ;
        ifstream in(*fileitr) ;
        while (in.get(c)) {
            if (c == '\n') { line_count++ ; }
        }
        nb_lines.push_back(line_count) ;
    }
}
```

En C++ ancien

```
int count_lines( const string & filename )  {
    ifstream in(filename) ;
    typedef istreambuf_iterator<char> ifiterator ;
    return count(ifiterator(in), ifiterator(), '\n') ;
}

vector<int> count_lines_in_files( const vector<string> & files )  {
    vector<int> results(files.size()) ;
    transform(files.cbegin(), files.cend(), results.begin(), count_lines) ;
    return results ;
}
```

En C++ moderne

```
auto count_lines( string const & filename ) -> int  {
    ifstream in(filename) ;
    using ifiterator = istreambuf_iterator<char>;
    return count(ifiterator(in), ifiterator(), '\n') ;
}

auto count_lines_in_files( vector<string> const & files )  -> vector<int> {
    vector<int> results(files.size()) ;
    transform(execution::par,files.cbegin(), files.cend(), results.begin(), count_lines) ;
    return results ;
}
```

En C++ futur

```
auto open_file( string const & filename )  -> ifstream {
    return ifstream(filename) ;
 }

auto count_lines( ifstream file )  -> int {
    using ifiterator = istreambuf_iterator<char> ;
    return count( ifiterator(in), ifiterator(), '\n') ;
}

auto count_lines_in_files( vector<string> const & files ) -> vector<int> {
    return files | transform(open_file) | transform(execution::par,count_lines) ;
}
```

Vectorisation SIMD

de la performance qui dort

Ecole de GIF, septembre 2019
David Chamont, LAL

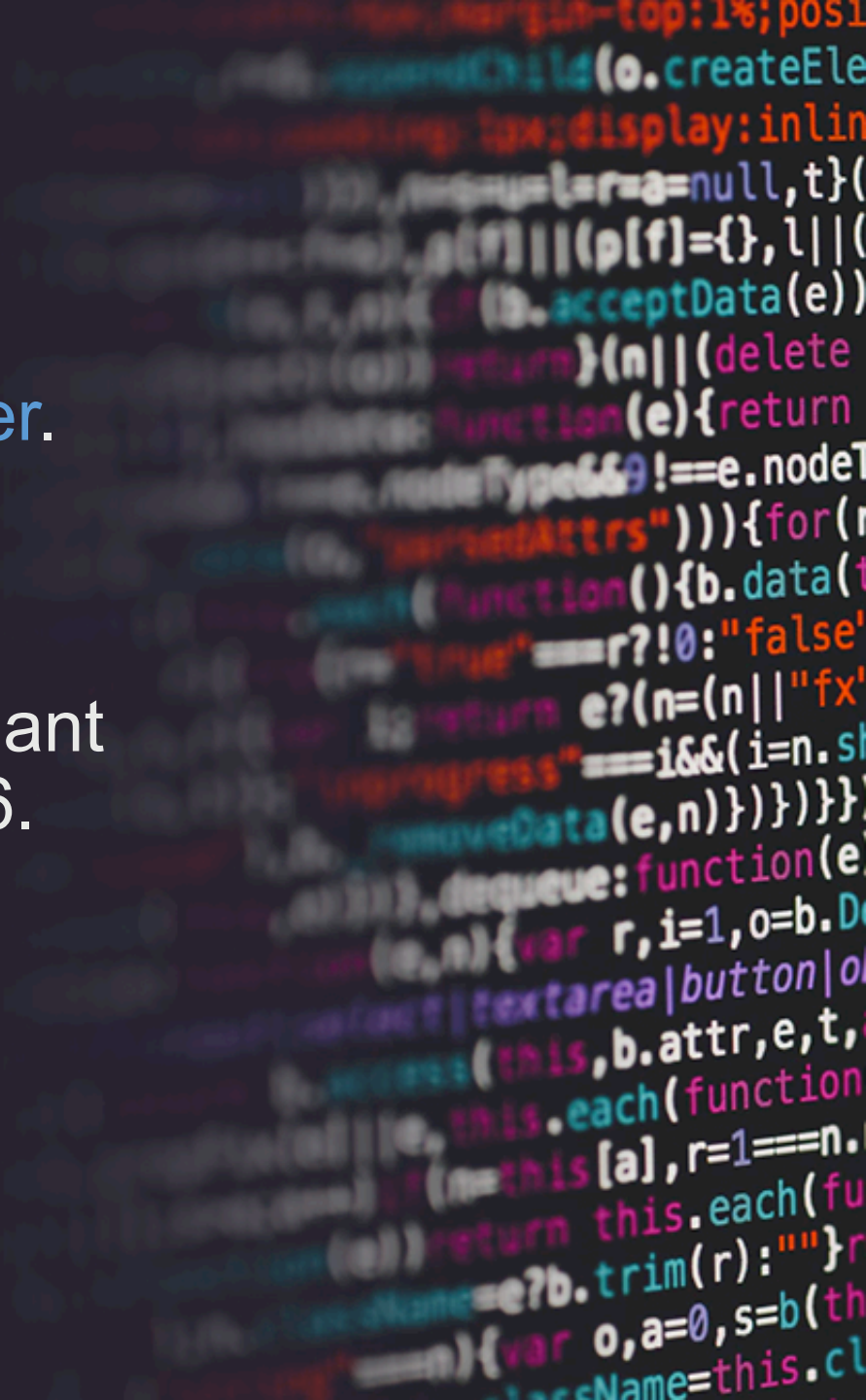
Pourquoi vectoriser ?

- Les processeurs possèdent depuis longtemps des registres vectoriels, qui tendent à s'allonger.
- Pour le prix d'une seule opération, vous pouvez en faire 4, 8, 16... selon votre matériel et la précision voulue.

➢ Pourquoi s'en priver ?

Comment l'obtenir ?

- En demandant au compilateur d'auto-vectoriser.
- A l’aide de directives OpenMP.
- A l’aide d’une bibliothèque (VC, Xsimd, ...).
- En programmant au plus bas niveau, en appelant explicitement des instructions intrinsèques x86.



Boost::SIMD

```
#include <boost/simd/include/functions/load.hpp>
#include <boost/simd/include/functions/store.hpp>
#include <boost/simd/include/functions/plus.hpp>
#include <boost/simd/include/functions/multiplies.hpp>

typedef simd::pack<double> pdouble ;
static const int psize = pdouble::static_size ;

.....
```

Boost::SIMD

```
.....

void compute_powers ( int n, double * xreal, double * ximag, double * yreal, double * yimag, int d )
 {
  int i = 0 ;
  while ( i < n )
   {
    // load an simd set of values
    pdouble pxreal = simd::load<pdouble>(xreal) ;
    pdouble pximag = simd::load<pdouble>(ximag) ;

    // Computation
    pdouble prreal(1.0), primag(0.0), ptmp(0.0) ;
    for ( int j=0 ; j < d ; j++ )
     {
      ptmp    = prreal*pxreal - primag*pximag ;
      primag  = prreal*pximag + primag*pxreal ;
      prreal  = ptmp ;
     }

    .....
```

Boost::SIMD

```
.....

   // store the result
   simd::store<pdouble>(prreal,yreal) ;
   simd::store<pdouble>(primag,yimag) ;

   // advance to the next simd vector
   i += psize ;
   xreal += psize ; ximag += psize ;
   yreal += psize ; yimag += psize ;
  }
}

.....
```

Boost::SIMD

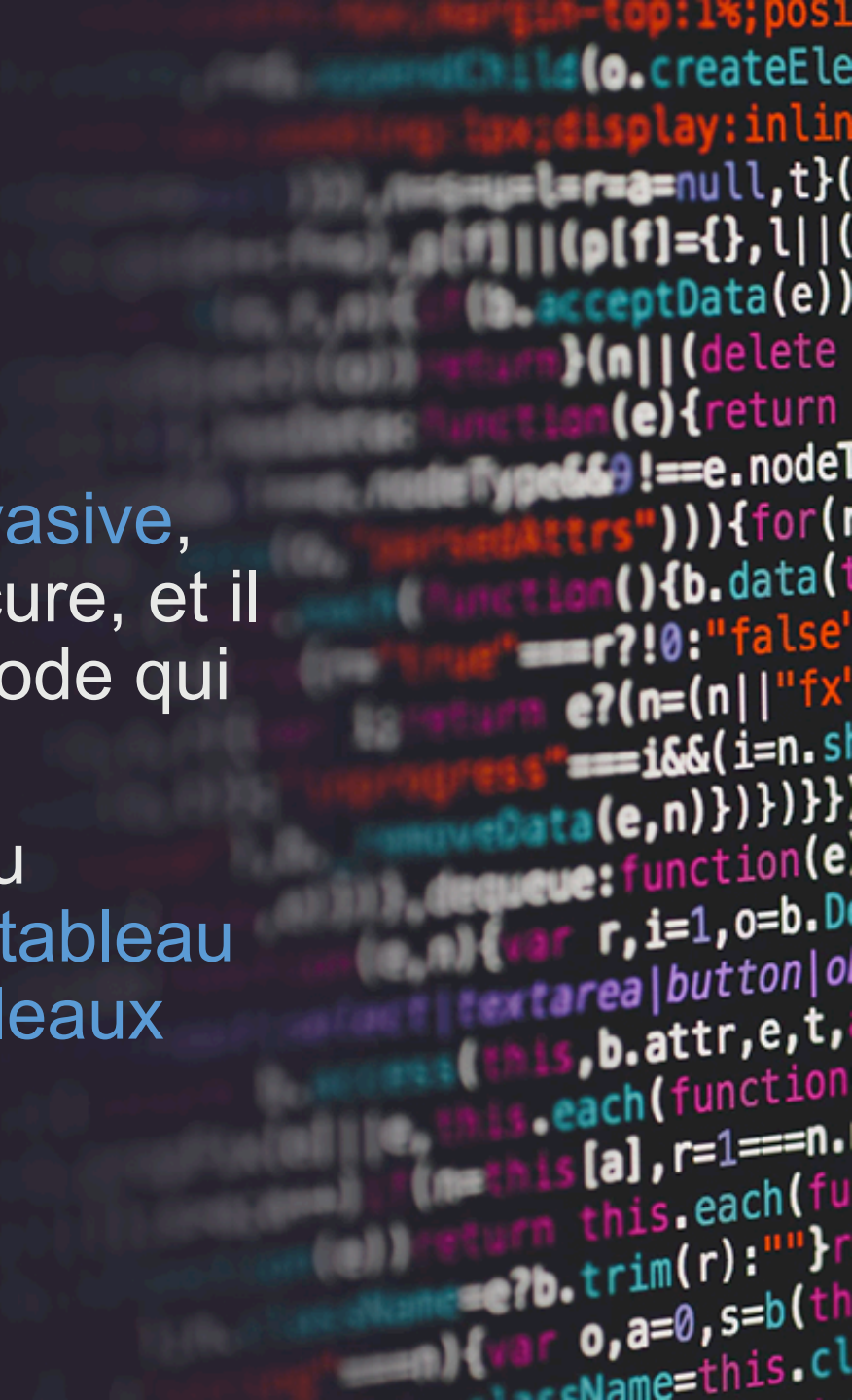
```
.....

// programme principal
int main( int argc, char * argv[] )
 {
  .....

  // prepare arrays
  double * const inputr = new double [dim] ;
  double * const inputi = new double [dim] ;
  double * const outputr = new double [dim] ;
  double * const outputi = new double [dim] ;

  // generate input
  .....

  .....
```

Boost::SIMD

```
.....

 // compute
 compute_powers(dim,inputr,inputi,outputr,outputi,degree) ;

 // post-process output
 .....

 // cleaning
 delete [] inputr ;
 delete [] inputi ;
 delete [] outputr ;
 delete [] outputi ;
 return 0 ;
}
```

Etat des lieux SIMD

- Aucune bibliothèque ne domine à présent.
- L'auto-vectorisation est l'approche la moins invasive, mais elle s'active ou non de façon parfois obscure, et il faut aider le compilateur en lui présentant un code qui s'y prête.
- La première étape est toujours de s'attaquer au modèle de données, et de remplacer les AoS (tableau d'objets) par des SoA (objet contenant des tableaux d'attributs).



Calcul déporté sur GPU

oui mais quand même

Ecole de GIF, septembre 2019
David Chamont, LAL

Pourquoi les GPUs ?

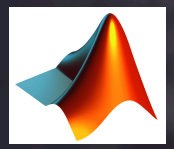
- Le meilleur rapport Flops/Watt
- En conjonction avec le déluge de données
  qui favorise le retour de l'apprentissage automatique

➢ trio gagnant GPU / AA / BigData

- Les GPUs envahissent les supercalculateurs qui font la course au TOP500 et au GREEN500.
- Les applications non-AA doivent s'en accomoder...

Comment les utiliser ?

- A travers une application déjà instrumentée
  - Matlab, Mathematica, ...
- A l'aide d'une bibliothèque haut niveau
  - CuBLAS, ArrayFire, TensorFlow, Kokkos, ...
- A l'aide de directives
  - OpenACC, OpenMP 4
- En programmant au plus bas niveau
  - CUDA
  - **OpenCL**, SyCL, ...



ArrayFire

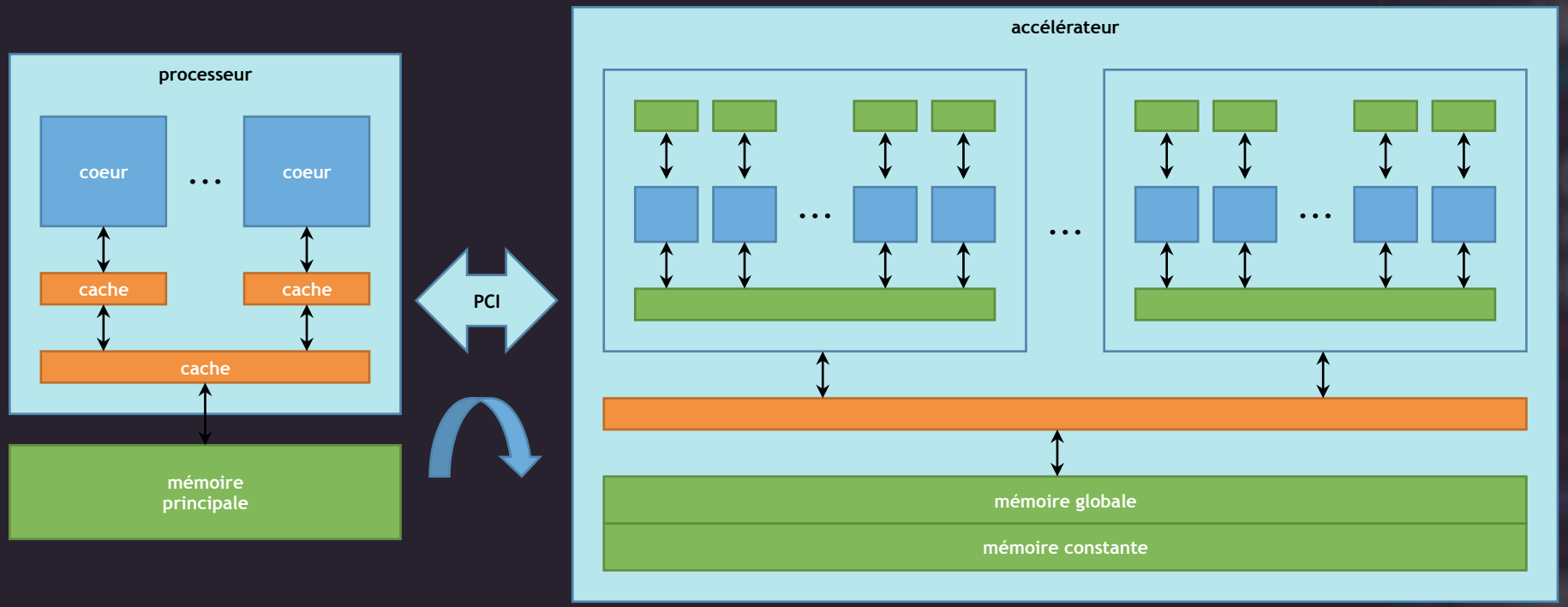
OpenACC
DIRECTIVES FOR ACCELERATORS

OpenMP

NVIDIA CUDA

OpenCL

Structure d'un GPU



OpenCL

```
#include <CL/opencl.h>

...

const char * KernelSource = "         \\n" \\
"  __kernel void square(              \\n" \\
"    __global float* input,           \\n" \\
"    __global float* output,          \\n" \\
"    const unsigned int count)        \\n" \\
"  {                                  \\n" \\
"    int i = get_global_id(0);        \\n" \\
"    if(i < count)                    \\n" \\
"      output[i] = input[i]*input[i]; \\n" \\
"  }                                  \\n" \\
"                                     \\n";

...
```

OpenCL

```
.....

int main(int argc, char** argv)
{
  // Host input data
  float data[DATA_SIZE];

  // Prepare kernel
  int err;
  cl_device_id device_id;
  err = clGetDeviceIDs(NULL,CL_DEVICE_TYPE_GPU,1,&device_id,NULL);
  cl_context context = clCreateContext(0,1,&device_id,NULL,NULL,&err);
  cl_command_queue queue = clCreateCommandQueue(context,device_id,0,&err);
  cl_program prog = clCreateProgramWithSource(context,1,(const char **)&KernelSource,...
  err = clBuildProgram(prog,0,NULL,NULL,NULL,NULL);
  cl_kernel kernel = clCreateKernel(prog,"square",&err);

  // Device io arrays
  cl_mem input = clCreateBuffer(context,CL_MEM_READ_ONLY,sizeof(float)*count,...
  cl_mem output = clCreateBuffer(context,CL_MEM_WRITE_ONLY,sizeof(float)*count,...

  .....
```

OpenCL

```
.....

// Write our data set into the input array in device memory
err = clEnqueueWriteBuffer(queue,input,CL_TRUE,0,sizeof(float)*count,data,0,...

// Set arguments to kernel
err  = clSetKernelArg(kernel,0,sizeof(cl_mem),&input);
err |= clSetKernelArg(kernel,1,sizeof(cl_mem),&output);
err |= clSetKernelArg(kernel,2,sizeof(unsigned int),&count);

// Queue kernel and wait for its execution end
size_t global = count;
err = clEnqueueNDRangeKernel(queue,kernel,1,NULL,&global,NULL,0,NULL,NULL);
clFinish(commands);

// Read back the results from the device to verify the output
float results[DATA_SIZE];
err = clEnqueueReadBuffer(queue,output,CL_TRUE,0,sizeof(float)*count,results,0,...

.....
```

OpenCL

```
.....

// Process results
.....

// Shutdown and cleanup
clReleaseMemObject(input);
clReleaseMemObject(output);
clReleaseProgram(prog);
clReleaseKernel(kernel);
clReleaseCommandQueue(commands);
clReleaseContext(context);
return 0 ;
}
```

OpenACC

```
.....

void compute_powers
 ( int n, double * xreal, double * ximag,
  double * restrict yreal, double * restrict yimag, int d
 )
 {
# pragma acc kernels loop copyin(xreal[0:n],ximag[0:n]) copyout(yreal[0:n],yimag[0:n])
  for ( int i=0 ; i<n ; ++i )
   {
    double rreal_tmp, rreal = 1.0, rimag = 0.0 ;
    for (int j=0; j < d; j++)
     {
      rreal_tmp = rreal*xreal[i] - rimag*ximag[i] ;
      rimag     = rreal*ximag[i] + rimag*xreal[i] ;
      rreal     = rreal_tmp ;
     }
    yreal[i] = rreal ; yimag[i] = rimag ;
   }
 }

.....
```

OpenACC

```
.....

int main ( int argc, char * argv[] )
 {
  .....

  // prepare arrays
  double * inputr = (double *)malloc(dim*sizeof(double)) ;
  double * inputi = (double *)malloc(dim*sizeof(double)) ;
  double * outputr = (double *)malloc(dim*sizeof(double)) ;
  double * outputi = (double *)malloc(dim*sizeof(double)) ;

  // generate input
  .....
```

OpenACC

```
.....

 // compute
 compute_powers(dim,inputr,inputi,outputr,outputi,degree) ;

 // process results
 .....

 // cleaning
 free(inputr) ;
 free(inputi) ;
 free(outputr) ;
 free(outputi) ;

 return 0 ;
}
```

Thrust

```
#include <thrust/host_vector.h>
#include <thrust/device_vector.h>
#include <thrust/transform.h>

.....

struct complex
 {
  double real ;
  double imag ;
 } ;

.....
```

Thrust

```
.....

struct compute_powers
 {
  public :
    compute_powers( int degree ) : degree_(degree) {}
    __host__ __device__
    complex operator()( const complex & c )
     {
      complex r ;
      r.real = 1.0 ; r.imag = 0.0 ;
      double real_tmp ;
      for( int  j=0 ; j < degree_ ; j++ ) {
        real_tmp = r.real * c.real - r.imag * c.imag ;
        r.imag = r.real * c.imag + r.imag * c.real ;
        r.real = real_tmp ;
      }
      return r ;
     }
  private :
    int degree_ ;
 } ;

.....
```

Thrust

```
.....

int main( int argc, char * argv[] )
 {
  .....

  // prepare arrays
  thrust::host_vector<complex> hinput(dim), houtput(dim) ;
  thrust::device_vector<complex> dinput(dim), doutput(dim) ;

  // prepare input
  .....

  // transfer and compute
  dinput = hinput ;
  thrust::transform(dinput.begin(),dinput.end(),doutput.begin(),compute_powers(degree)) ;
  houtput = doutput ;

   // process output
  .....

  return 0 ;
 }
```

Etat des lieux GPU

- Le meilleur rapport Flops/Watt... si le GPU tourne à plein régime.
- Il faut
  - l'alimenter en données (et ces copies ont un coût)
  - qu'il y ait beaucoup de calculs à faire pour chaque donnée
  - gérer le partage entre tous les coeurs de la carte mère
  - que ces coeurs sachent quoi faire en attendant les résultats
- La programmation est délicate, mais l'effort paiera quoi qu'il arrive, même si vous ne tournez pas finalement le programme sur GPU.

Données massives

après Hadoop, **Spark**

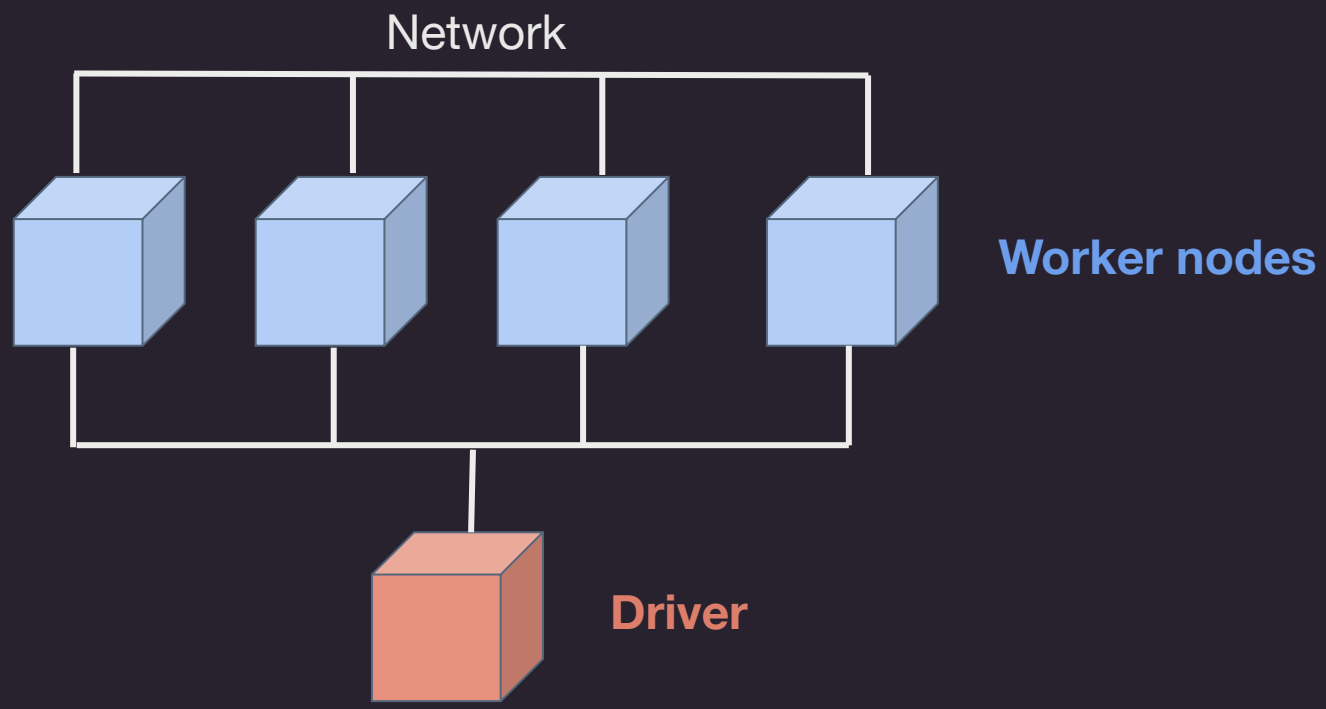
(remerciements à Julien Peloton)

Ecole de GIF, septembre 2019
David Chamont, LAL

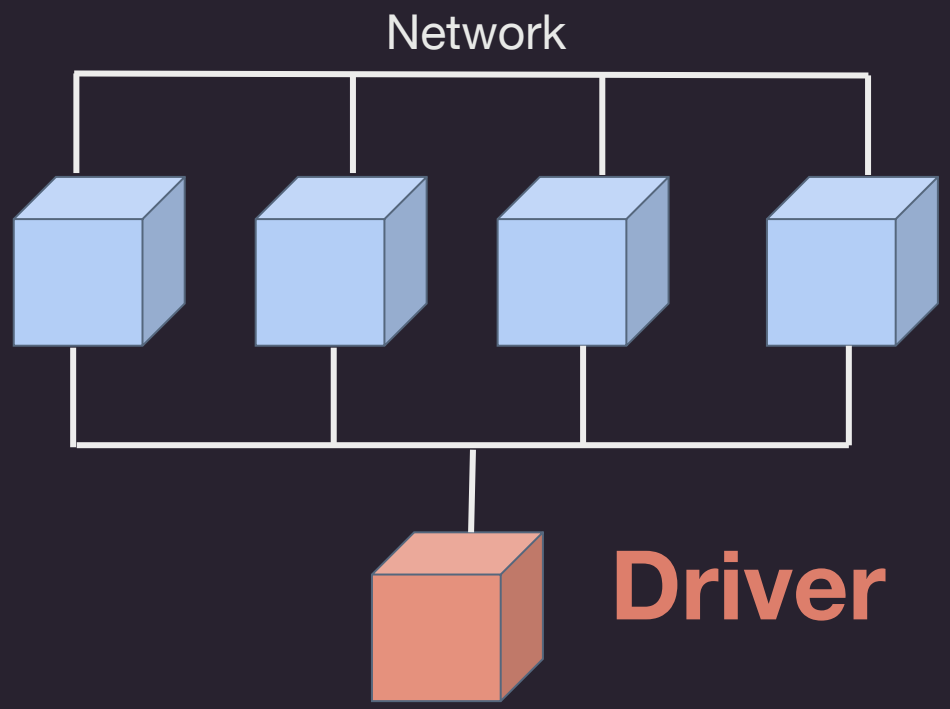
Apache Spark

- A distributed computing framework.
- Come from: Google MapReduce (2004), Apache Hadoop (2006).
  - started in 2009 (UC Berkeley/AMPLab),
  - moved to the Apache foundation in 2013,
  - v1 (2014), v2 (2016), v3 (exp. 2019).
- License: open source, mainly Apache-2.0.
- Used by (literally) more than 1000 companies… Insane!
- Use Scala (& Java & Python & R).

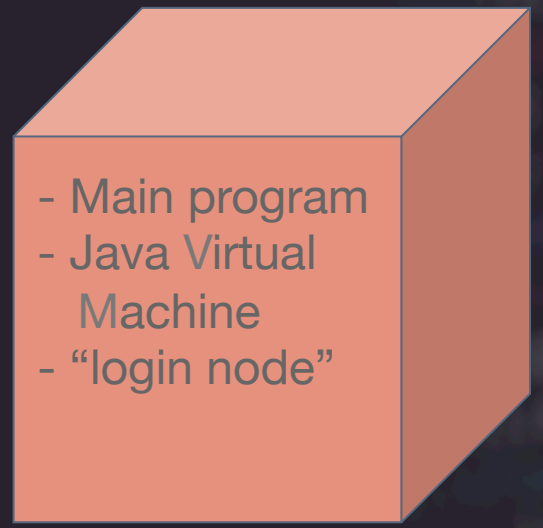
Spark distributed computing



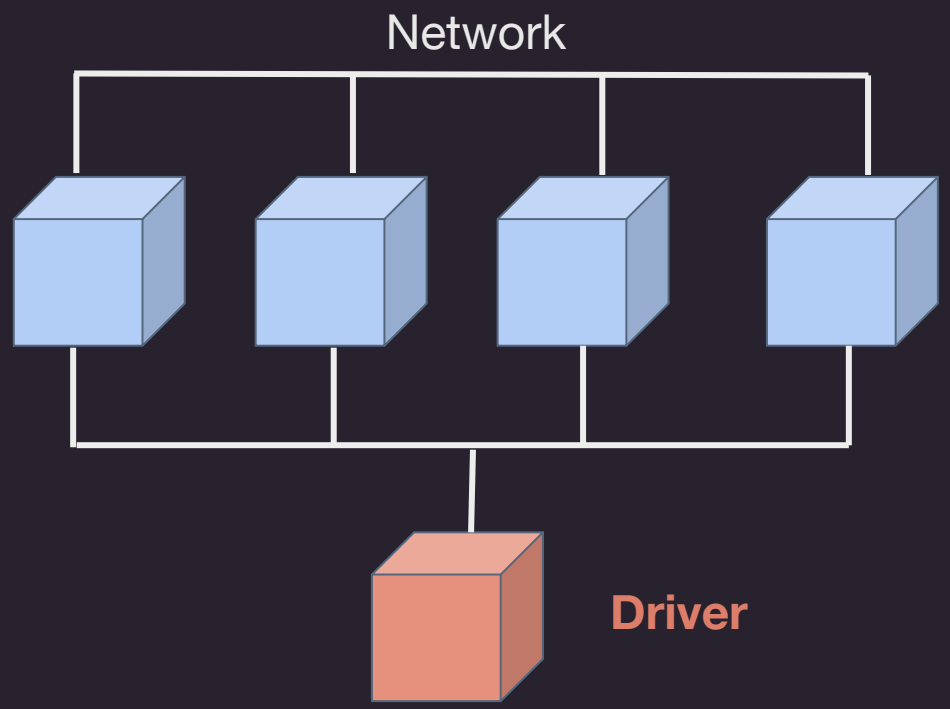
Spark distributed computing



Worker nodes



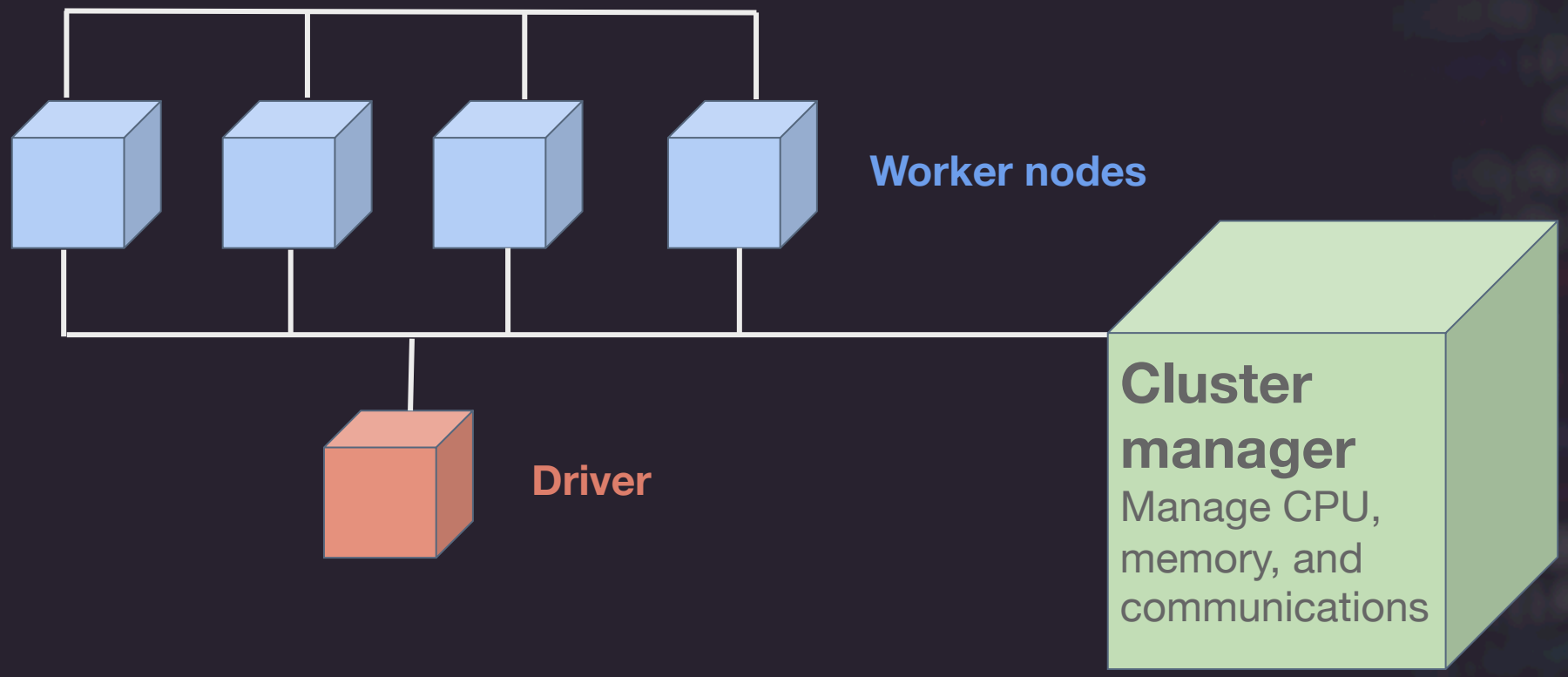
Spark distributed computing



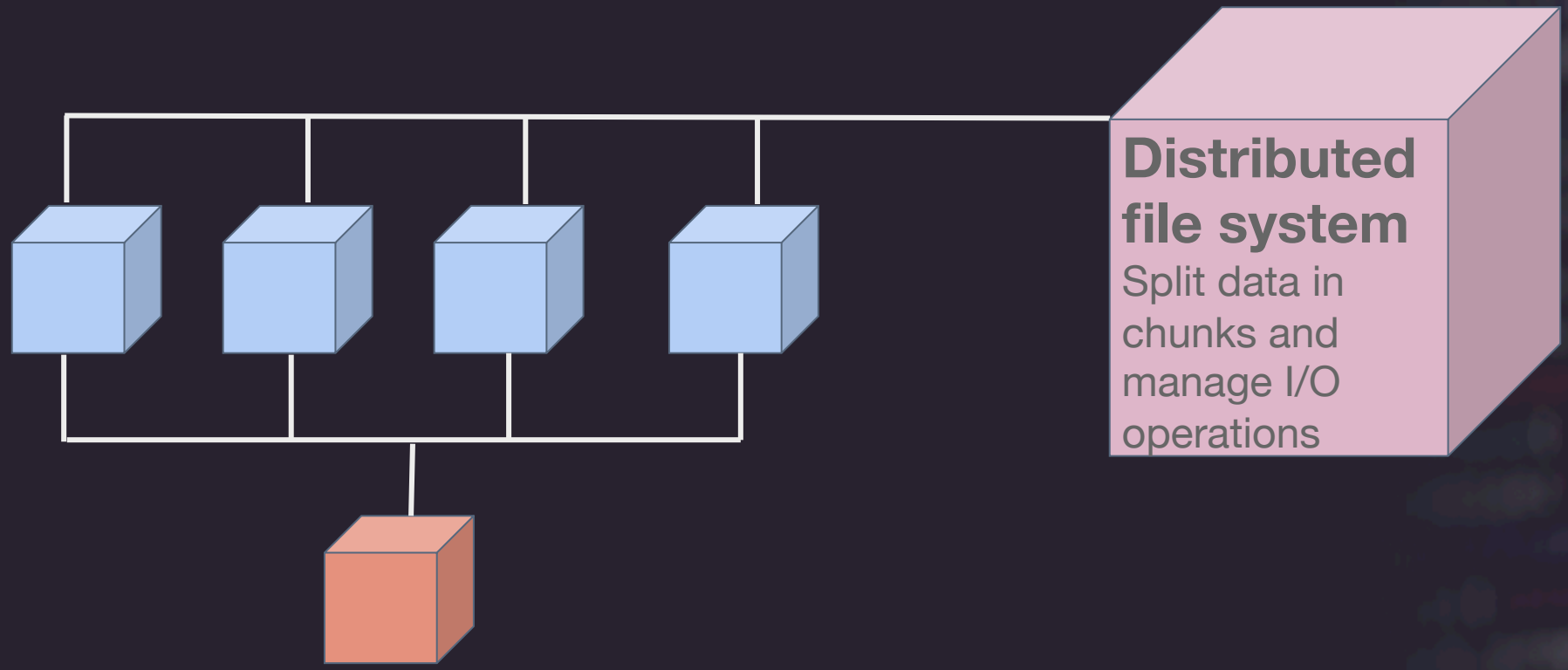
Worker nodes

- Contain executors (JVM)
- Multi-cores
- Run tasks

Spark distributed computing



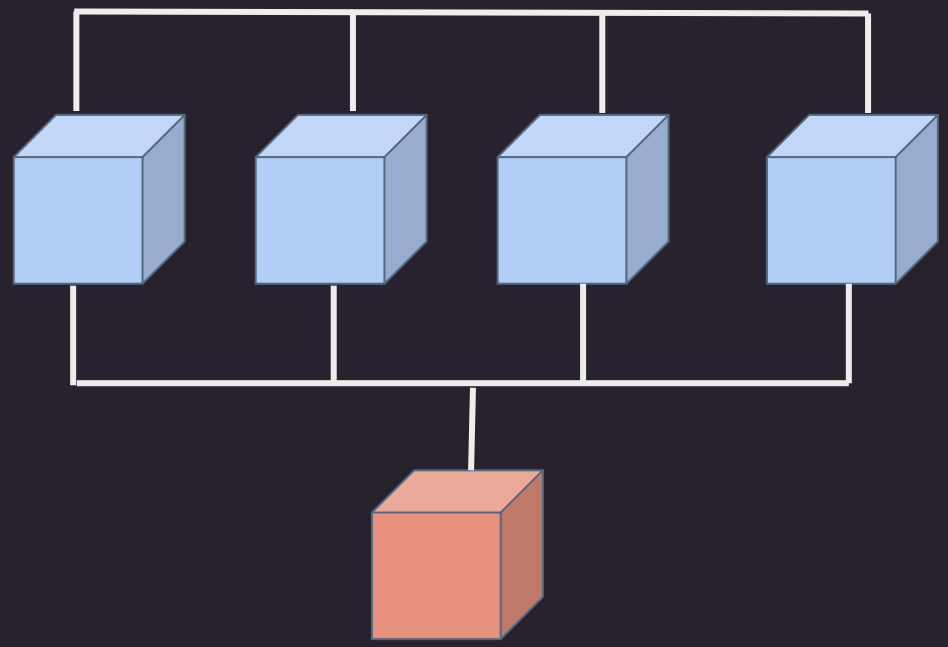
Spark distributed computing



Distributed file system

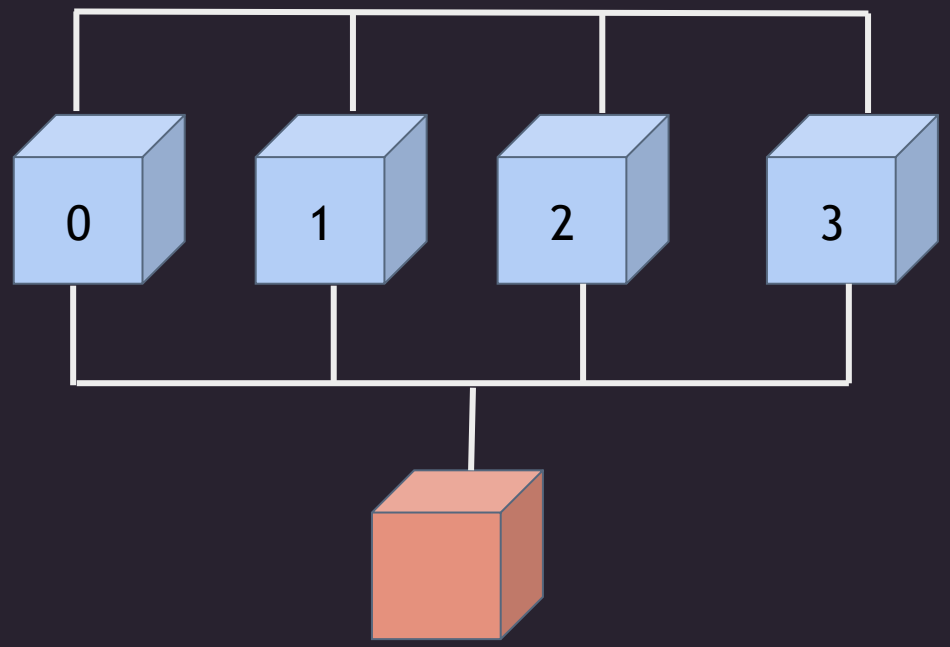
Split data in chunks and manage I/O operations

Spark DataFrames



x	y	z	radius
1.0	1.0	1.0	0.8
1.0	1.0	1.0	0.5
3.0	1.0	1.0	0.8
3.0	1.0	1.0	0.5
3.0	3.0	1.0	0.8
3.0	3.0	1.0	0.5
1.0	3.0	1.0	0.8
1.0	3.0	1.0	0.5
1.0	1.0	3.0	0.8

Spark DataFrames



Worker 0

x	y	z	radius
1.0	1.0	1.0	0.8
1.0	1.0	1.0	0.5

Worker 1

3.0	1.0	1.0	0.8
3.0	1.0	1.0	0.5
3.0	3.0	1.0	0.8

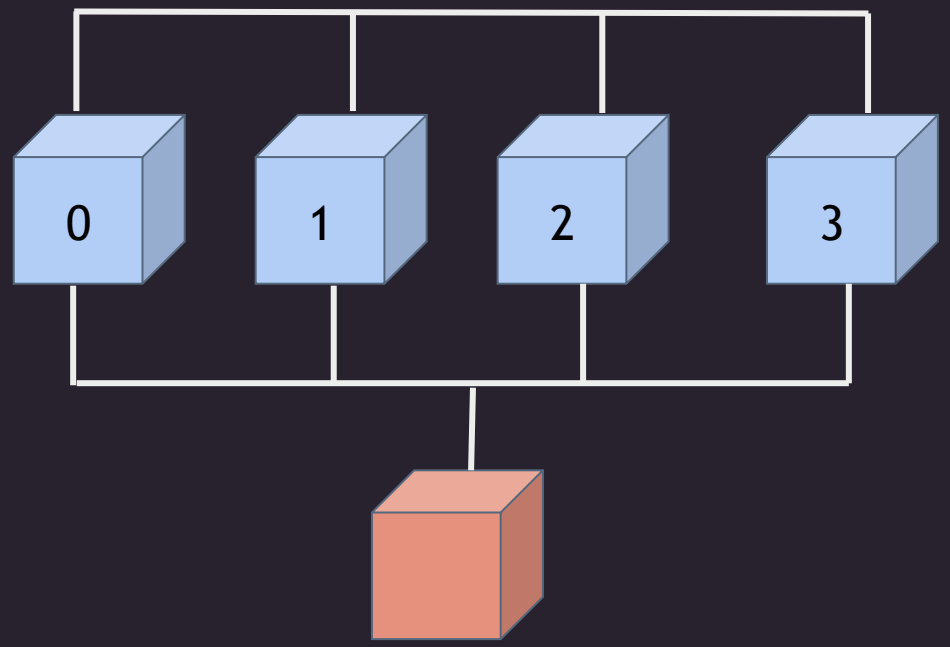
Worker 2

3.0	3.0	1.0	0.5
1.0	3.0	1.0	0.8

Worker 3

1.0	3.0	1.0	0.5
1.0	1.0	3.0	0.8
1.0	1.0	3.0	0.5
3.0	1.0	3.0	0.8

Spark is easy




```
df = spark.read\
    .format("...")\
    .option("...")\
    .load("path/to/file")
```

Distribution
Load balancing
DAG optimization
+ ...

Quelques liens

- Apache Spark research papers: https://spark.apache.org/research.html
- Apache Spark basics doc: https://spark.apache.org/docs/latest/quick-start.html
- AstroLab Software: https://github.com/astrolabsoftware

Utilisation de conteneurs

premier pas avec docker

Ecole de GIF, septembre 2019
David Chamont, LAL

Docker en quelques mots

- Permet d'exécuter du code linux à l'intérieur d'un conteneur isolé. Chaque conteneur embarque sa variante de linux et ses propres fichiers, mais ils partagent tous le même noyau, ce qui les rend beaucoup plus légers qu'une machine virtuelle.
- S'installe d'une commande ou d'un clic sur Linux, MacOS et Windows 10 → portabilité !
- S'attendre à quelques difficultés si vous voulez faire tourner une application avec interface graphique requérant X11.

Docker pull

- Le Docker Hub met à disposition quelques millions d'images prêtes à servir : https://hub.docker.com/
- Pour s'y retrouver, Un système de labels permet d'identifier les plus officielles et les plus sûres.
- La plupart des grands éditeurs de logiciel libre y mettent à disposition les différentes versions de leurs produits
- Exemple

```
docker pull ubuntu:18.04
```

Docker run

- Créer un conteneur à partir d'une image,
  et y lancer un shell interactif :

  ```
  docker run –it ubuntu:18.04 bash
  ```

- Compiler puis exécuter ponctuellement un programme

  ```
  docker run –it -v $PWD:/work ubuntu:18.04 g++ monprog.cpp
  ```

  ```
  docker run –it -v $PWD:/work ubuntu:18.04 ./a.out
  ```

- Lancer un serveur web en tâche de fond

  ```
  docker run –d –p 80:80 -v $PWD/html:/usr/share/nginx/html nginx
  ```

Docker build

- A l'aide d'une recette (Dockerfile)

```
FROM ubuntu:latest
RUN apt-get update && apt-get install -y python3
WORKDIR /home/www
CMD python3 /home/src/monwebserver.py
ADD index.html /home/www/index.html
ADD mon-webserver.py /home/src/monwebserver.py
```

- on peut construire sa propre image

```
docker build –tag=chamont/webserver .
```

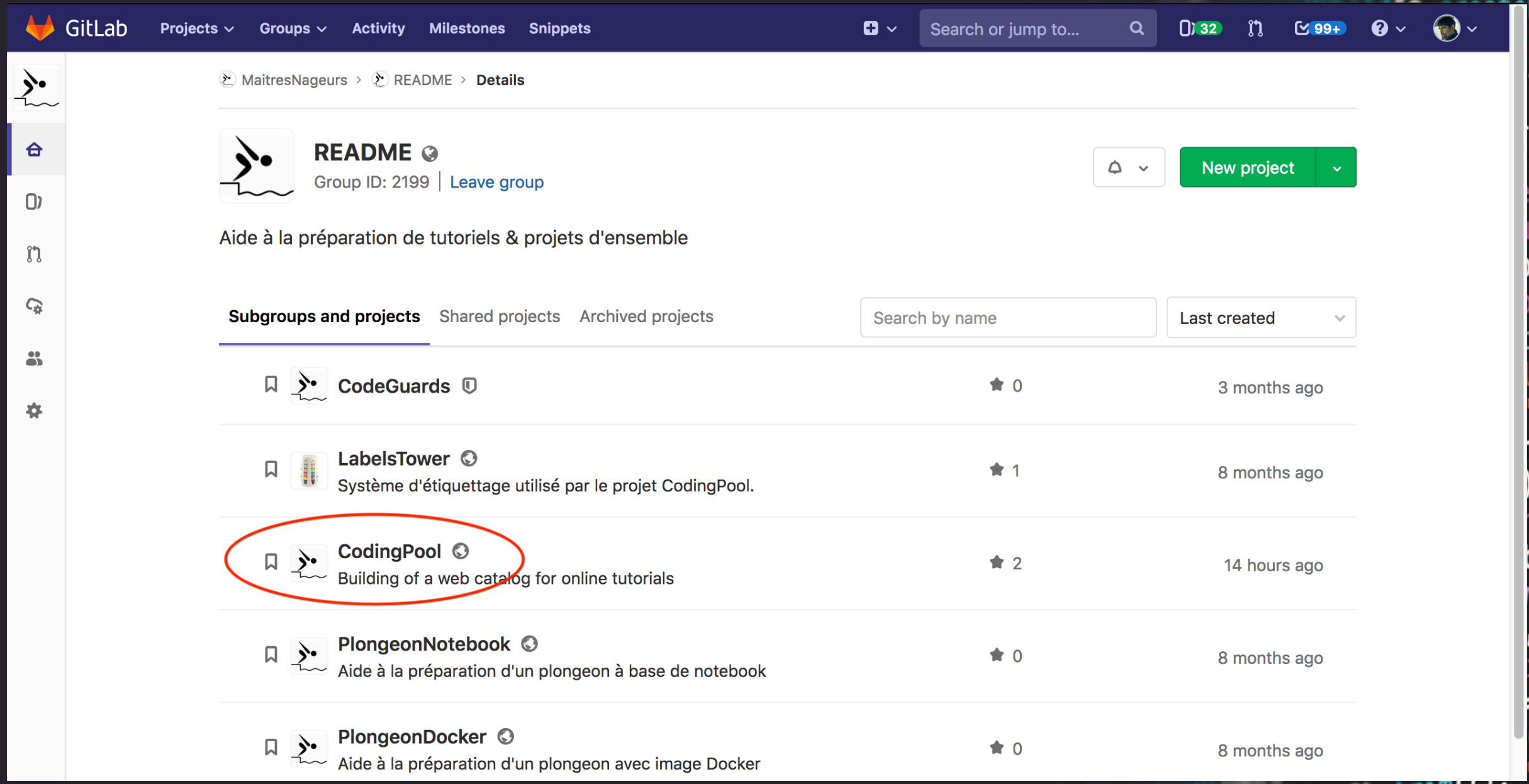
- et la partager via le hub

```
docker login ; docker push chamont/webserver .
```

Développement collaboratif

exemple idéalisé avec Gitlab

Ecole de GIF, septembre 2019
David Chamont, LAL


GitLab
Projects
Groups
Activity
Milestones
Snippets
Search or jump to...
32
99+
MaitresNageurs › README › Details
README
Group ID: 2199 | Leave group
New project
Aide à la préparation de tutoriels & projets d'ensemble
Subgroups and projects
Shared projects
Archived projects
Search by name
Last created
CodeGuards
0
3 months ago
LabelsTower
Système d'étiquettage utilisé par le projet CodingPool.
1
8 months ago
CodingPool
Building of a web catalog for online tutorials
2
14 hours ago
PlongeonNotebook
Aide à la préparation d'un plongeon à base de notebook
0
8 months ago
PlongeonDocker
Aide à la préparation d'un plongeon avec image Docker
0
8 months ago

GitLab Projects Groups Activity Milestones Snippets Search or jump to... 32 99+

MaitresNageurs › README › CodingPool › Details

CodingPool

Project ID: 4407

Unstar 2 Fork 3 Clone

Add license 218 Commits 2 Branches 1 Tag 41.6 MB Files

Building of a web catalog for online tutorials

Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

Learn more in the Auto DevOps documentation

Enable in settings

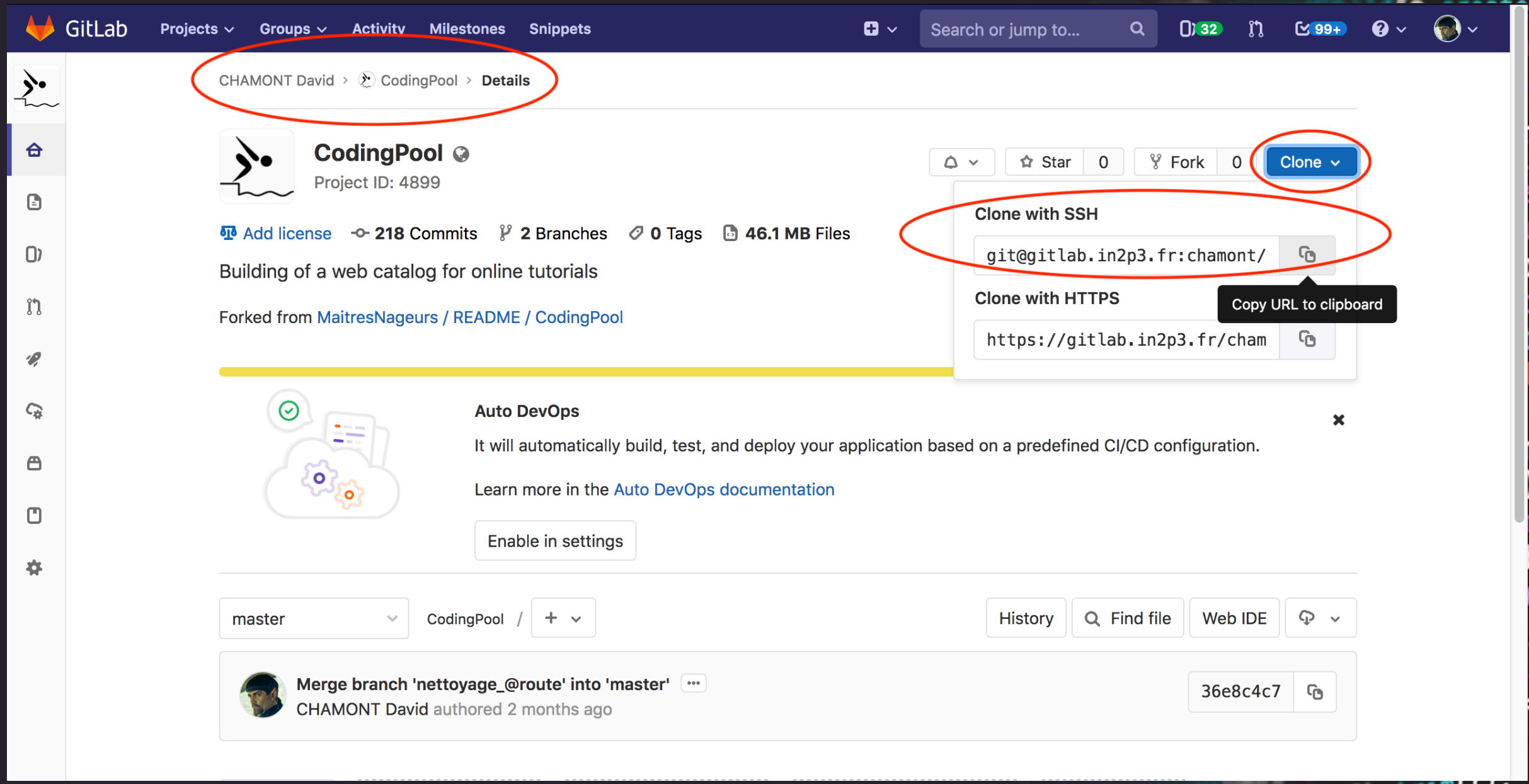
master CodingPool / + History Find file Web IDE

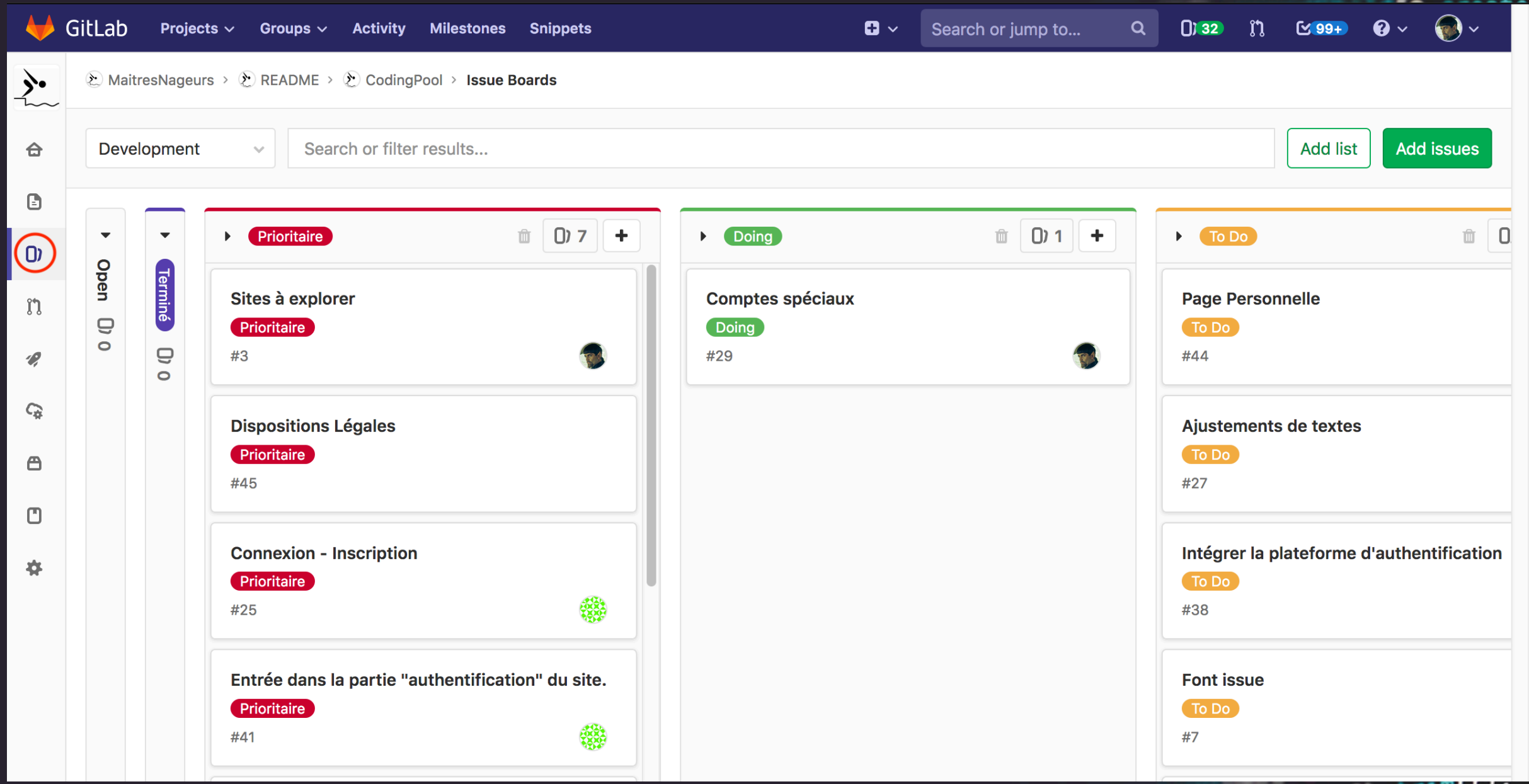
Merge branch 'nettoyage_@route' into 'master'

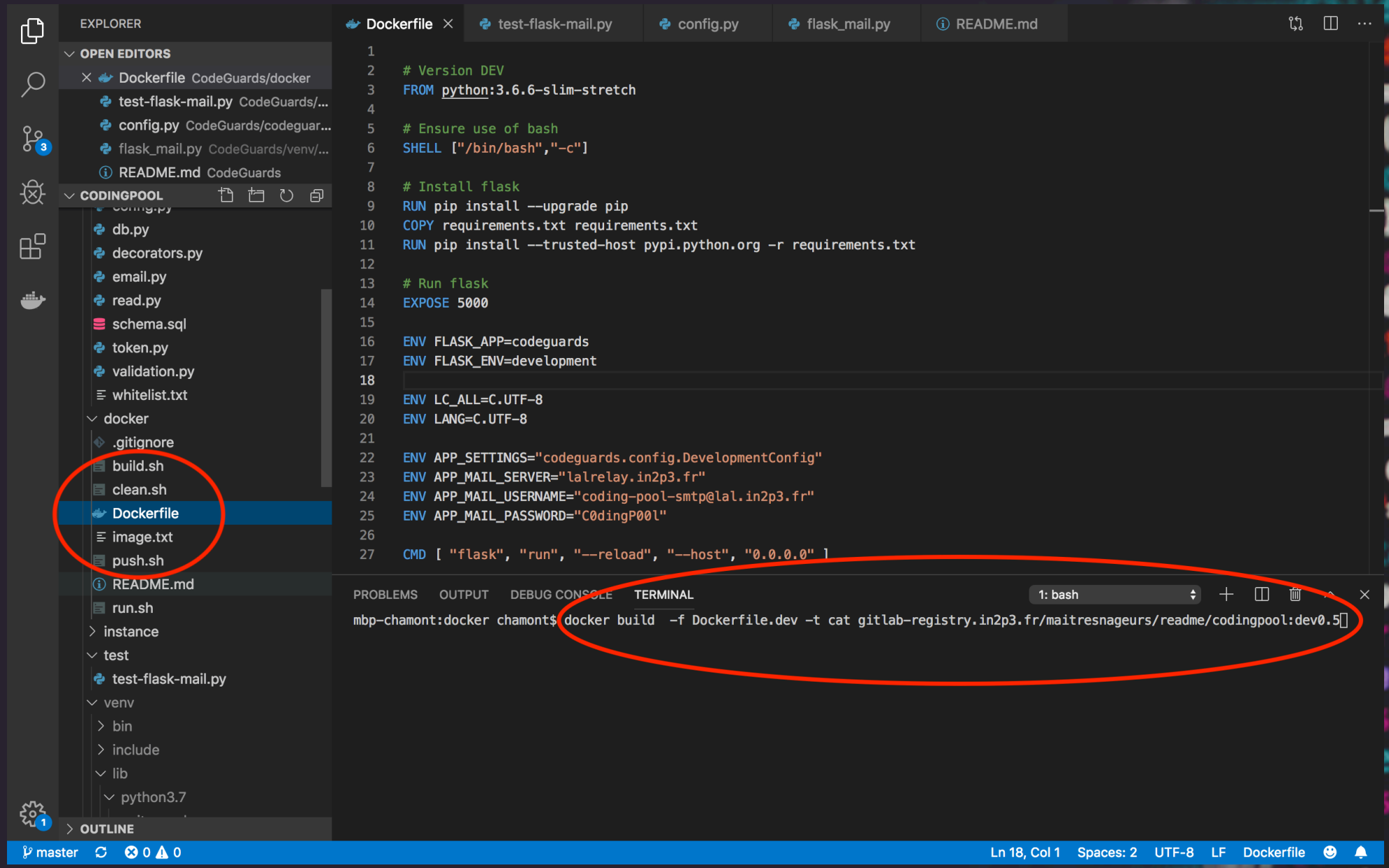
CHAMONT David authored 2 months ago

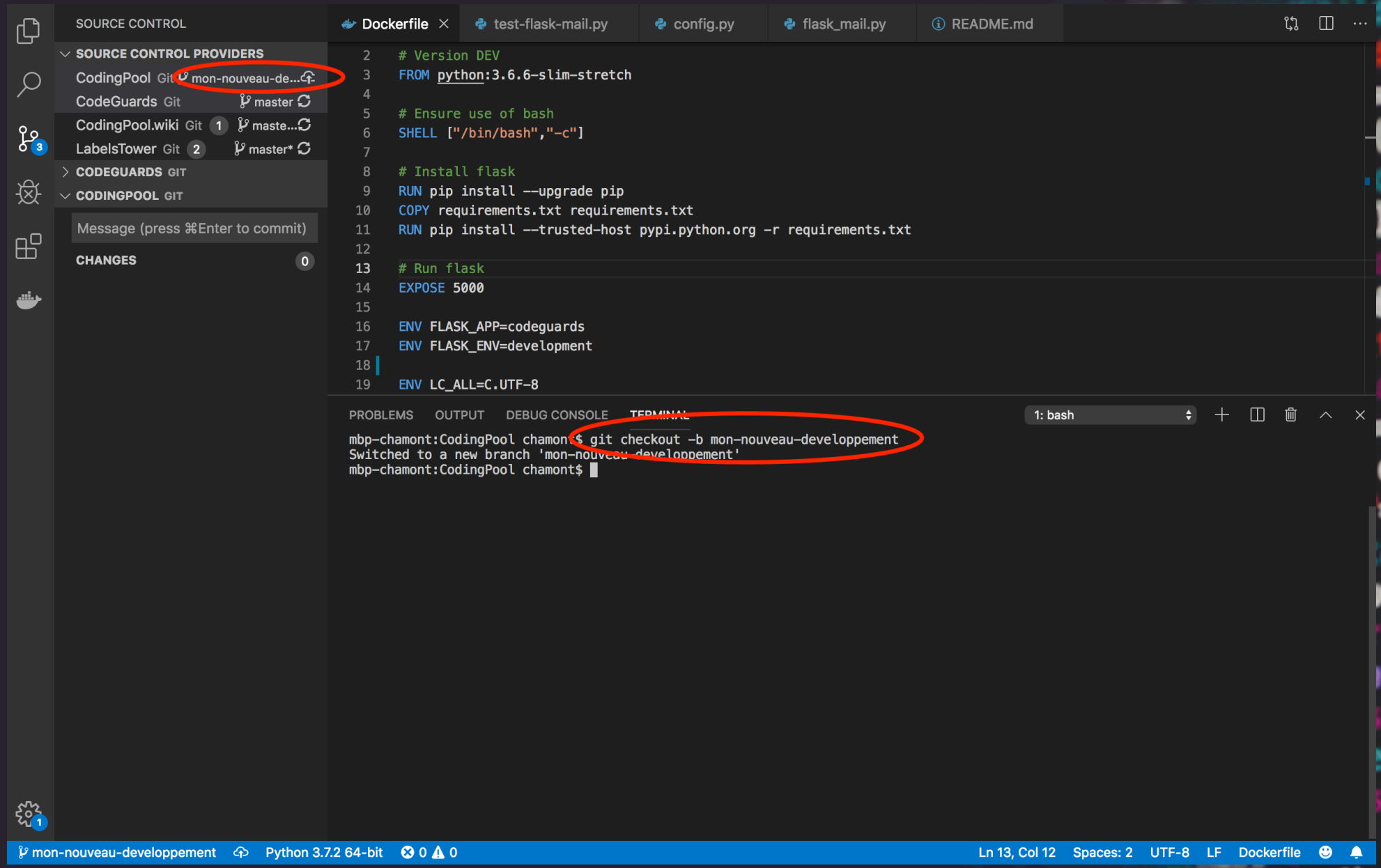
36e8c4c7

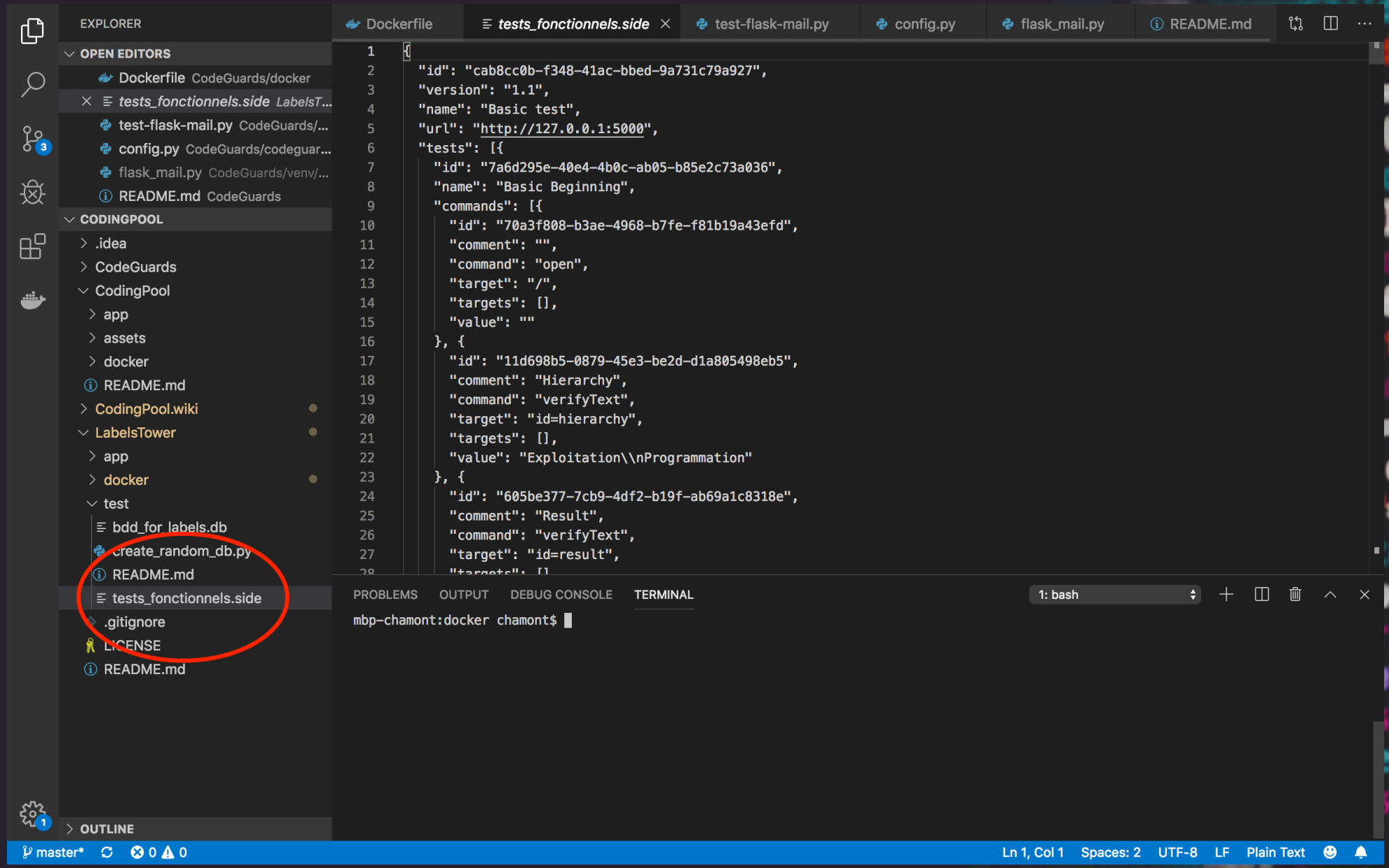
README Add CHANGELOG Add CONTRIBUTING Add Kubernetes cluster Set up CI/CD

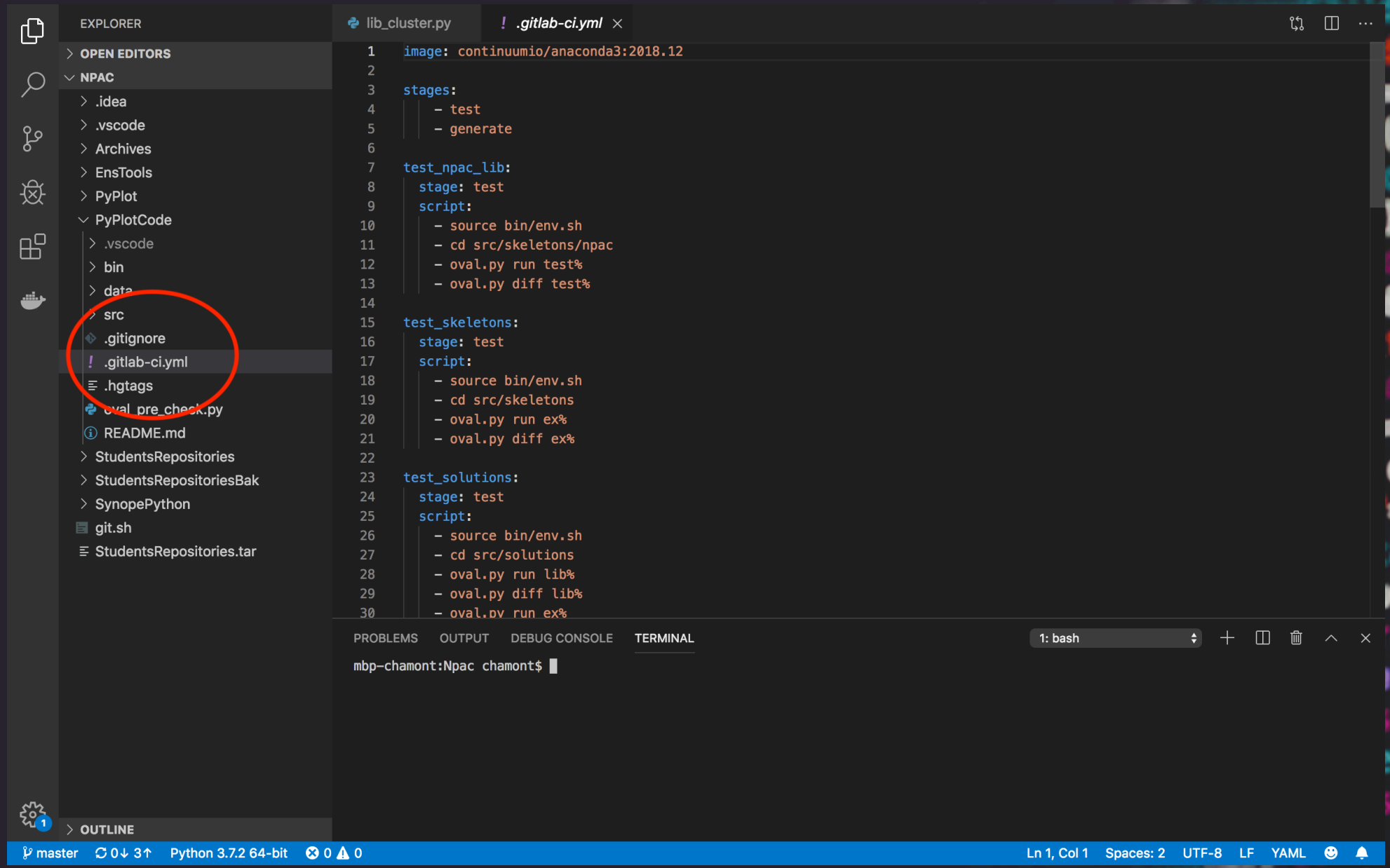

GitLab
Projects
Groups
Activity
Milestones
Snippets
Search or jump to...
CHAMONT David › CodingPool › Details
CodingPool
Project ID: 4899
Star 0
Fork 0
Clone
Clone with SSH
git@gitlab.in2p3.fr:chamont/
Copy URL to clipboard
Clone with HTTPS
https://gitlab.in2p3.fr/cham
Add license
218 Commits
2 Branches
0 Tags
46.1 MB Files
Building of a web catalog for online tutorials
Forked from MaitresNageurs / README / CodingPool
Auto DevOps
It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.
Learn more in the Auto DevOps documentation
Enable in settings
master
CodingPool /
History
Find file
Web IDE
Merge branch 'nettoyage_@route' into 'master'
CHAMONT David authored 2 months ago
36e8c4c7

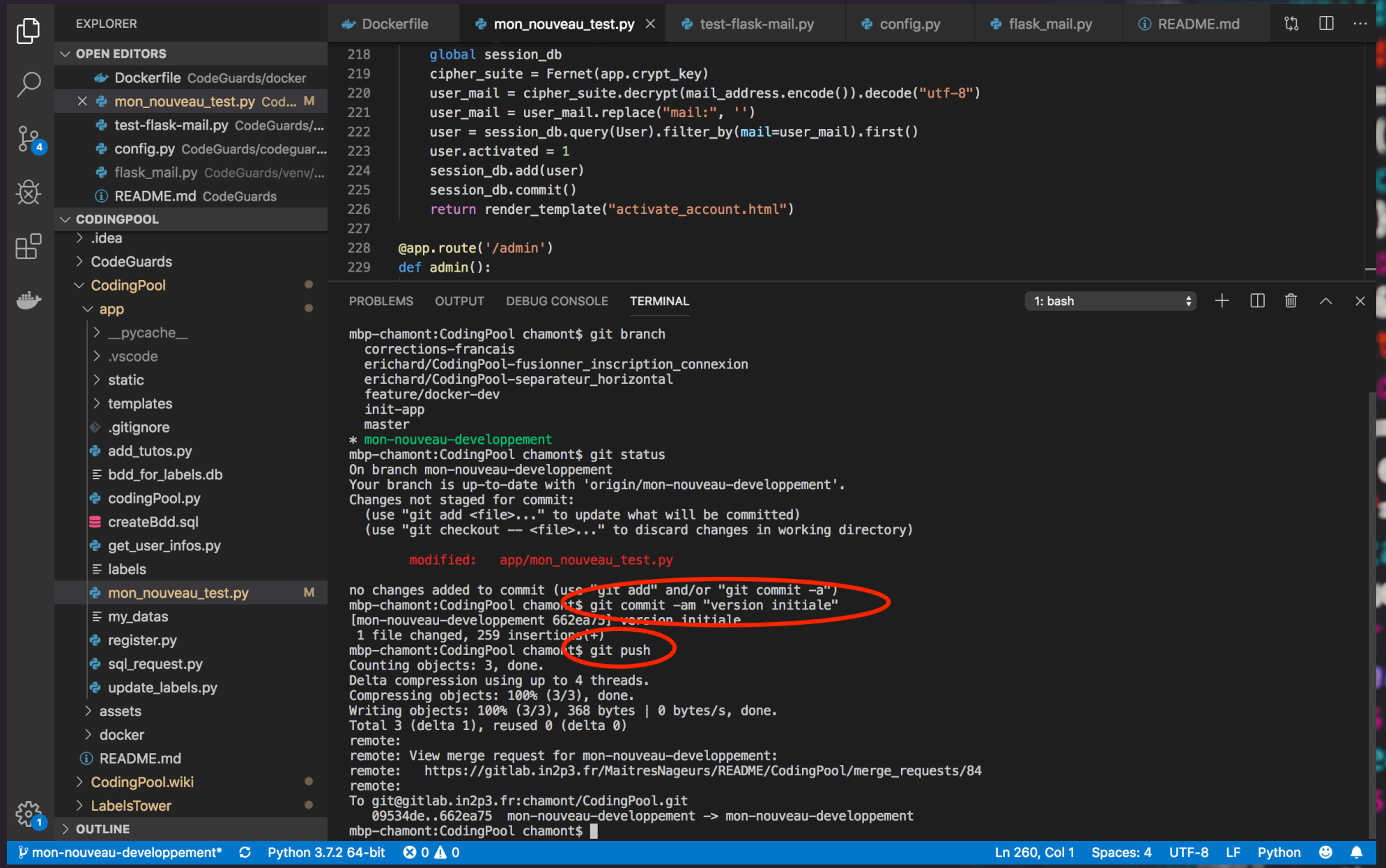

GitLab
Projects
Groups
Activity
Milestones
Snippets
Search or jump to...
32
99+
MaitresNageurs
README
CodingPool
Issue Boards
Development
Search or filter results...
Add list
Add issues
Open
0
Terminé
0
Prioritaire
7
Sites à explorer
Prioritaire
#3
Dispositions Légales
Prioritaire
#45
Connexion - Inscription
Prioritaire
#25
Entrée dans la partie "authentification" du site.
Prioritaire
#41
Doing
1
Comptes spéciaux
Doing
#29
To Do
Page Personnelle
To Do
#44
Ajustements de textes
To Do
#27
Intégrer la plateforme d'authentification
To Do
#38
Font issue
To Do
#7

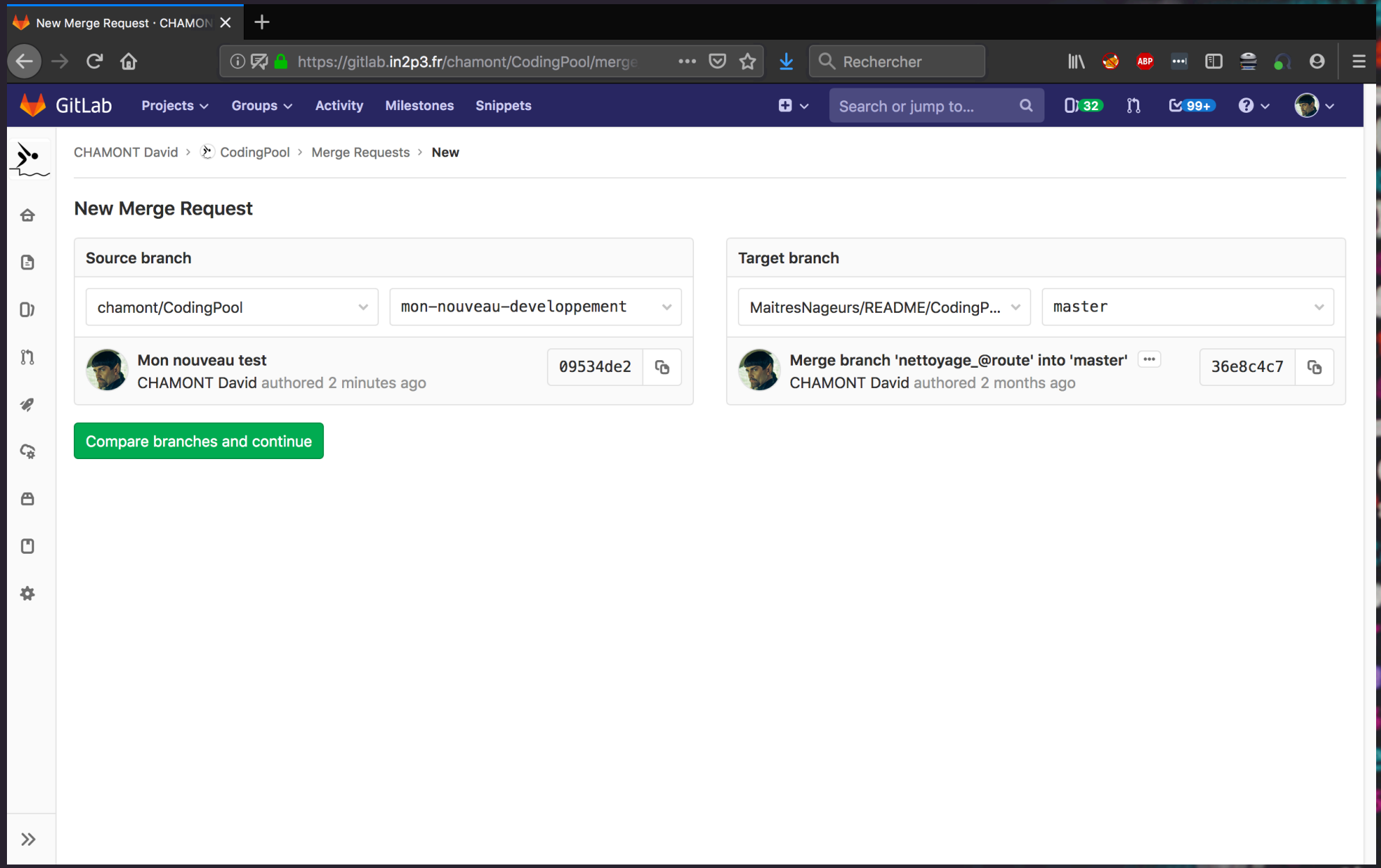

EXPLORER
OPEN EDITORS
Dockerfile CodeGuards/docker
test-flask-mail.py CodeGuards/...
config.py CodeGuards/codeguar...
flask_mail.py CodeGuards/venv/...
README.md CodeGuards
CODINGPOOL
db.py
decorators.py
email.py
read.py
schema.sql
token.py
validation.py
whitelist.txt
docker
.gitignore
build.sh
clean.sh
Dockerfile
image.txt
push.sh
README.md
run.sh
instance
test
test-flask-mail.py
venv
bin
include
lib
python3.7
OUTLINE
Dockerfile
test-flask-mail.py
config.py
flask_mail.py
README.md
Version DEV
FROM python:3.6.6-slim-stretch
Ensure use of bash
SHELL ["/bin/bash","-c"]
Install flask
RUN pip install --upgrade pip
COPY requirements.txt requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt
Run flask
EXPOSE 5000
ENV FLASK_APP=codeguards
ENV FLASK_ENV=development
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8
ENV APP_SETTINGS="codeguards.config.DevelopmentConfig"
ENV APP_MAIL_SERVER="lalrelay.in2p3.fr"
ENV APP_MAIL_USERNAME="coding-pool-smtp@lal.in2p3.fr"
ENV APP_MAIL_PASSWORD="C0dingP00l"
CMD ["flask", "run", "--reload", "--host", "0.0.0.0"]
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: bash
mbp-chamont:docker chamont$ docker build -f Dockerfile.dev -t cat gitlab-registry.in2p3.fr/maitresnageurs/readme/codingpool:dev0.5
master
0 0
Ln 18, Col 1
Spaces: 2
UTF-8
LF
Dockerfile

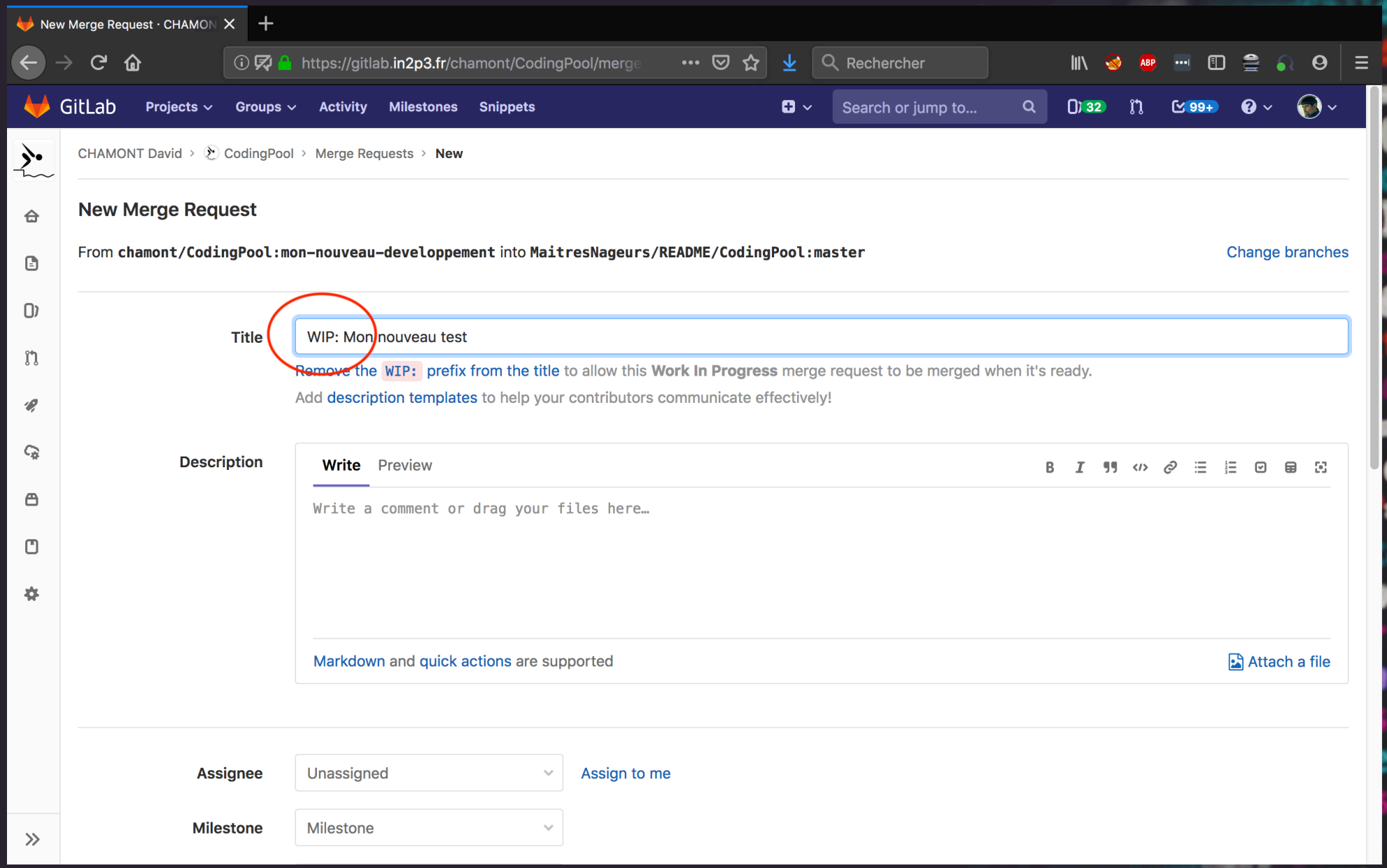

SOURCE CONTROL
SOURCE CONTROL PROVIDERS
CodingPool Git mon-nouveau-de...
CodeGuards Git master
CodingPool.wiki Git 1 maste...
LabelsTower Git 2 master*
CODEGUARDS GIT
CODINGPOOL GIT
Message (press ⌘Enter to commit)
CHANGES 0
Dockerfile
test-flask-mail.py
config.py
flask_mail.py
README.md
Version DEV
FROM python:3.6.6-slim-stretch
Ensure use of bash
SHELL ["/bin/bash","-c"]
Install flask
RUN pip install --upgrade pip
COPY requirements.txt requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt
Run flask
EXPOSE 5000
ENV FLASK_APP=codeguards
ENV FLASK_ENV=development
ENV LC_ALL=C.UTF-8
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: bash
mbp-chamont:CodingPool chamont$ git checkout -b mon-nouveau-developpement
Switched to a new branch 'mon-nouveau-developpement'
mbp-chamont:CodingPool chamont$
mon-nouveau-developpement
Python 3.7.2 64-bit
0 0
Ln 13, Col 12
Spaces: 2
UTF-8
LF
Dockerfile

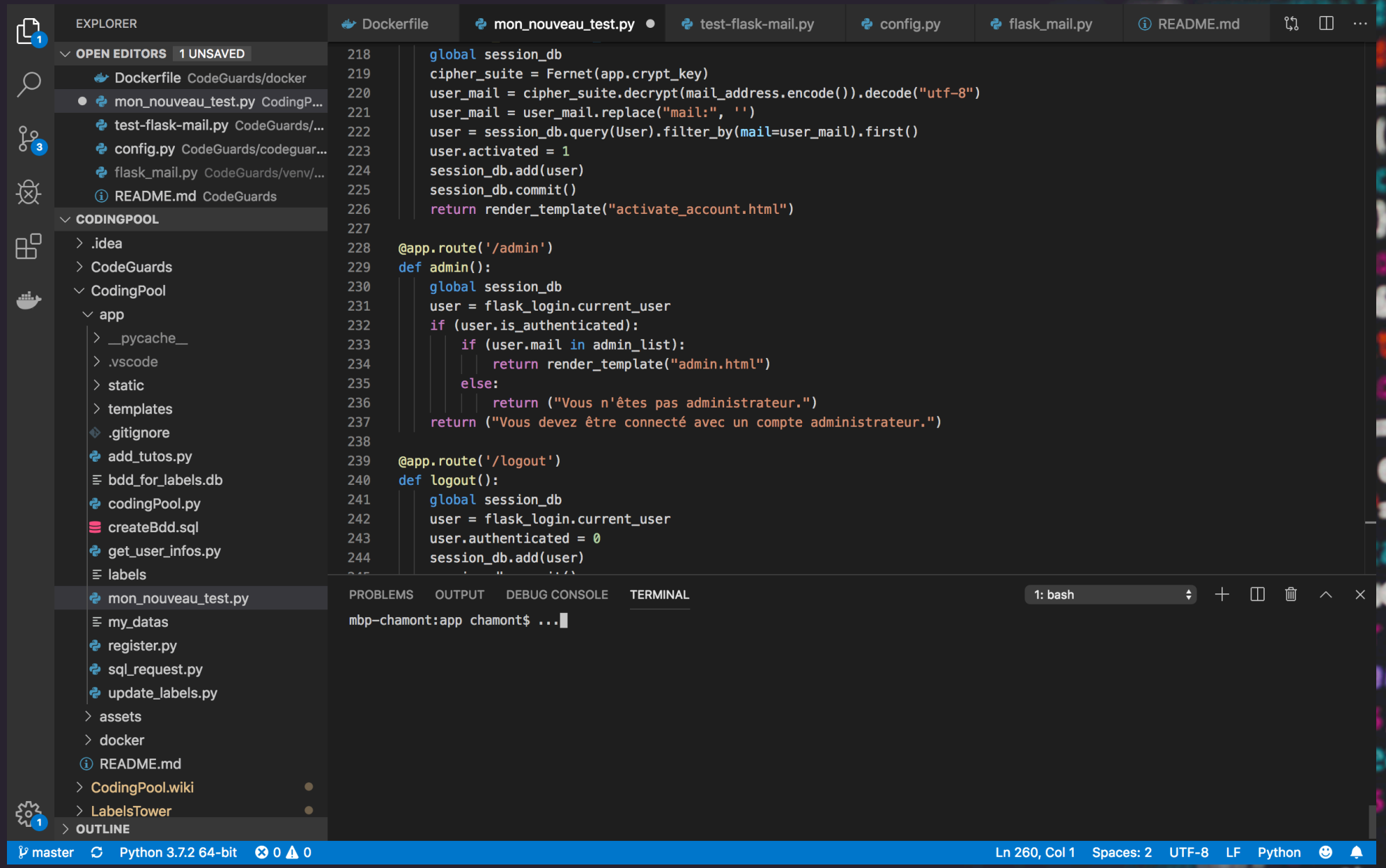

EXPLORER
OPEN EDITORS
Dockerfile CodeGuards/docker
tests_fonctionnels.side LabelsT...
test-flask-mail.py CodeGuards/...
config.py CodeGuards/codeguar...
flask_mail.py CodeGuards/venv/...
README.md CodeGuards
CODINGPOOL
.idea
CodeGuards
CodingPool
app
assets
docker
README.md
CodingPool.wiki
LabelsTower
app
docker
test
bdd_for_labels.db
create_random_db.py
README.md
tests_fonctionnels.side
.gitignore
LICENSE
README.md
OUTLINE
Dockerfile
tests_fonctionnels.side
test-flask-mail.py
config.py
flask_mail.py
README.md
{
"id": "cab8cc0b-f348-41ac-bbed-9a731c79a927",
"version": "1.1",
"name": "Basic test",
"url": "http://127.0.0.1:5000",
"tests": [{
"id": "7a6d295e-40e4-4b0c-ab05-b85e2c73a036",
"name": "Basic Beginning",
"commands": [{
"id": "70a3f808-b3ae-4968-b7fe-f81b19a43efd",
"comment": "",
"command": "open",
"target": "/",
"targets": [],
"value": ""
}, {
"id": "11d698b5-0879-45e3-be2d-d1a805498eb5",
"comment": "Hierarchy",
"command": "verifyText",
"target": "id=hierarchy",
"targets": [],
"value": "Exploitation\\nProgrammation"
}, {
"id": "605be377-7cb9-4df2-b19f-ab69a1c8318e",
"comment": "Result",
"command": "verifyText",
"target": "id=result",
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: bash
mbp-chamont:docker chamont$
master*
0 0
Ln 1, Col 1
Spaces: 2
UTF-8
LF
Plain Text

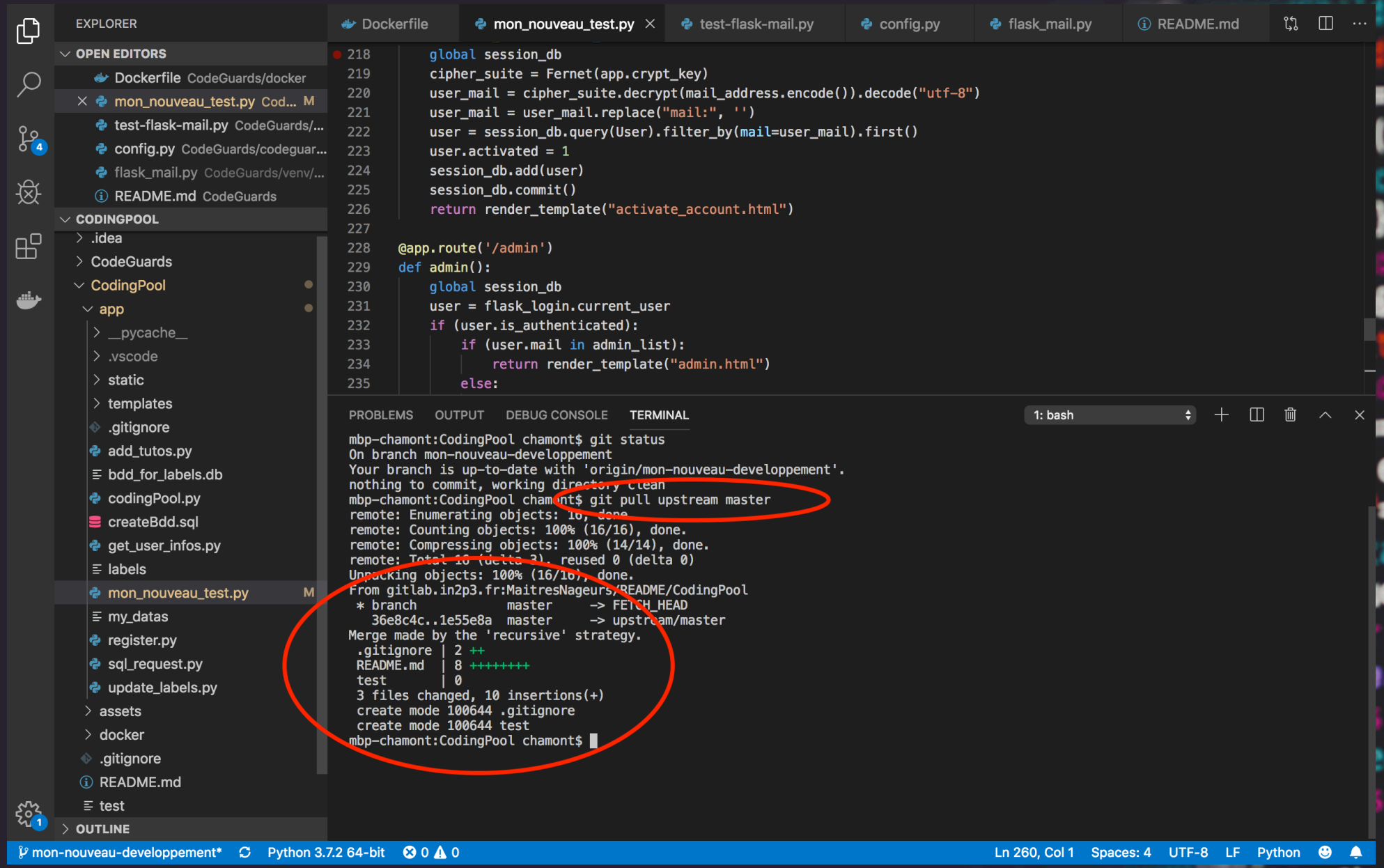

EXPLORER
OPEN EDITORS
NPAC
.idea
.vscode
Archives
EnsTools
PyPlot
PyPlotCode
.vscode
bin
data
src
.gitignore
.gitlab-ci.yml
.hgtags
oval_pre_check.py
README.md
StudentsRepositories
StudentsRepositoriesBak
SynopePython
git.sh
StudentsRepositories.tar
OUTLINE
lib_cluster.py
.gitlab-ci.yml
image: continuumio/anaconda3:2018.12
stages:
- test
- generate
test_npac_lib:
stage: test
script:
- source bin/env.sh
- cd src/skeletons/npac
- oval.py run test%
- oval.py diff test%
test_skeletons:
stage: test
script:
- source bin/env.sh
- cd src/skeletons
- oval.py run ex%
- oval.py diff ex%
test_solutions:
stage: test
script:
- source bin/env.sh
- cd src/solutions
- oval.py run lib%
- oval.py diff lib%
- oval.py run ex%
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: bash
mbp-chamont:Npac chamont$
master
0↓ 3↑
Python 3.7.2 64-bit
0 0
Ln 1, Col 1
Spaces: 2
UTF-8
LF
YAML

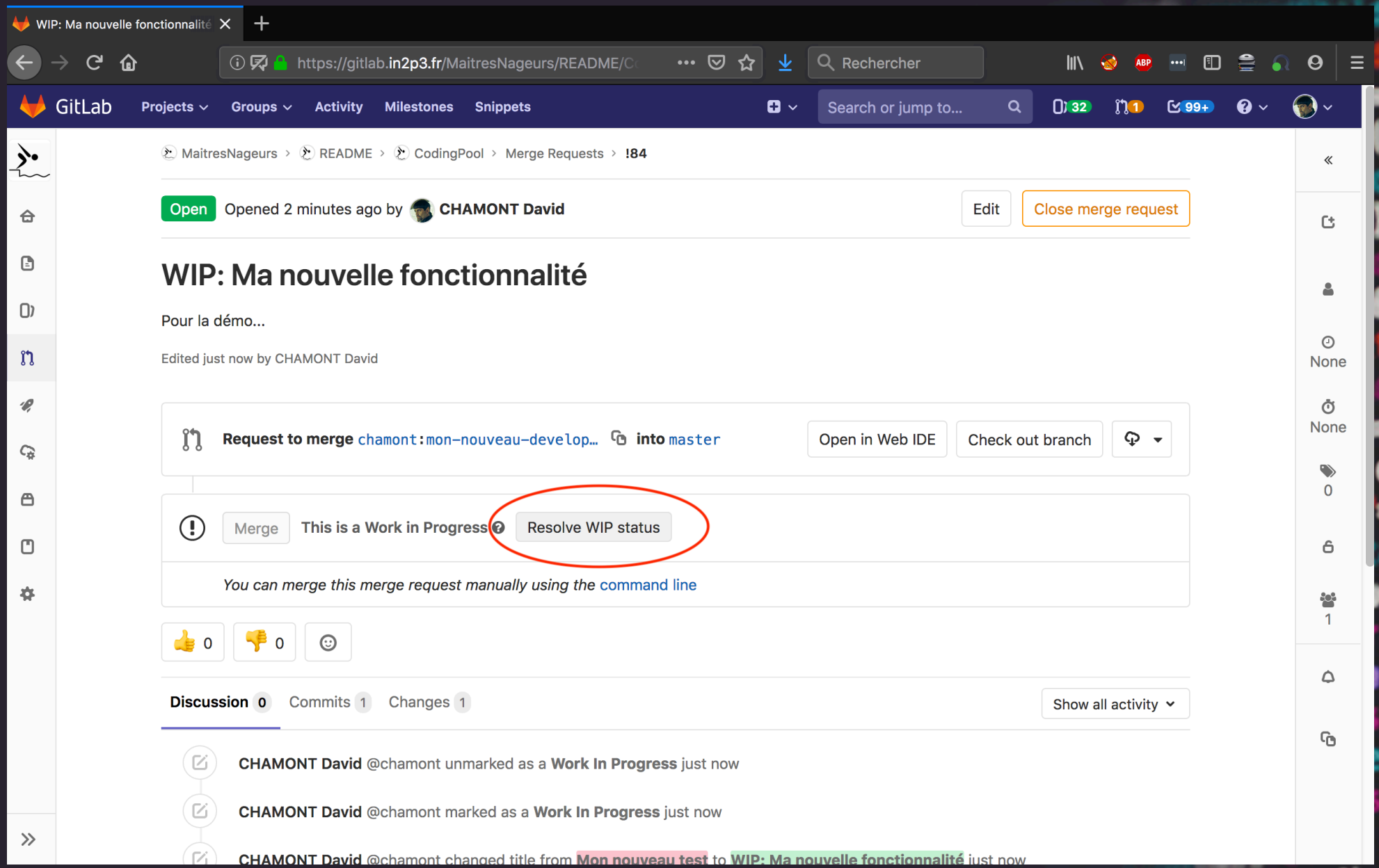

EXPLORER
OPEN EDITORS
Dockerfile CodeGuards/docker
mon_nouveau_test.py Cod... M
test-flask-mail.py CodeGuards/...
config.py CodeGuards/codeguar...
flask_mail.py CodeGuards/venv/...
README.md CodeGuards
CODINGPOOL
.idea
CodeGuards
CodingPool
app
__pycache__
.vscode
static
templates
.gitignore
add_tutos.py
bdd_for_labels.db
codingPool.py
createBdd.sql
get_user_infos.py
labels
mon_nouveau_test.py M
my_datas
register.py
sql_request.py
update_labels.py
assets
docker
README.md
CodingPool.wiki
LabelsTower
OUTLINE
Dockerfile
mon_nouveau_test.py
test-flask-mail.py
config.py
flask_mail.py
README.md
218 global session_db
219 cipher_suite = Fernet(app.crypt_key)
220 user_mail = cipher_suite.decrypt(mail_address.encode()).decode("utf-8")
221 user_mail = user_mail.replace("mail:", '')
222 user = session_db.query(User).filter_by(mail=user_mail).first()
223 user.activated = 1
224 session_db.add(user)
225 session_db.commit()
226 return render_template("activate_account.html")
227
228 @app.route('/admin')
229 def admin():
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: bash
mbp-chamont:CodingPool chamont$ git branch
corrections-francais
erichard/CodingPool-fusionner_inscription_connexion
erichard/CodingPool-separateur_horizontal
feature/docker-dev
init-app
master
* mon-nouveau-developpement
mbp-chamont:CodingPool chamont$ git status
On branch mon-nouveau-developpement
Your branch is up-to-date with 'origin/mon-nouveau-developpement'.
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)
modified: app/mon_nouveau_test.py
no changes added to commit (use "git add" and/or "git commit -a")
mbp-chamont:CodingPool chamont$ git commit -am "version initiale"
[mon-nouveau-developpement 662ea75] version initiale
1 file changed, 259 insertions(+)
mbp-chamont:CodingPool chamont$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 368 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote:
remote: View merge request for mon-nouveau-developpement:
remote: https://gitlab.in2p3.fr/MaitresNageurs/README/CodingPool/merge_requests/84
remote:
To git@gitlab.in2p3.fr:chamont/CodingPool.git
09534de..662ea75 mon-nouveau-developpement -> mon-nouveau-developpement
mbp-chamont:CodingPool chamont$
mon-nouveau-developpement*
Python 3.7.2 64-bit
0 0
Ln 260, Col 1
Spaces: 4
UTF-8
LF
Python

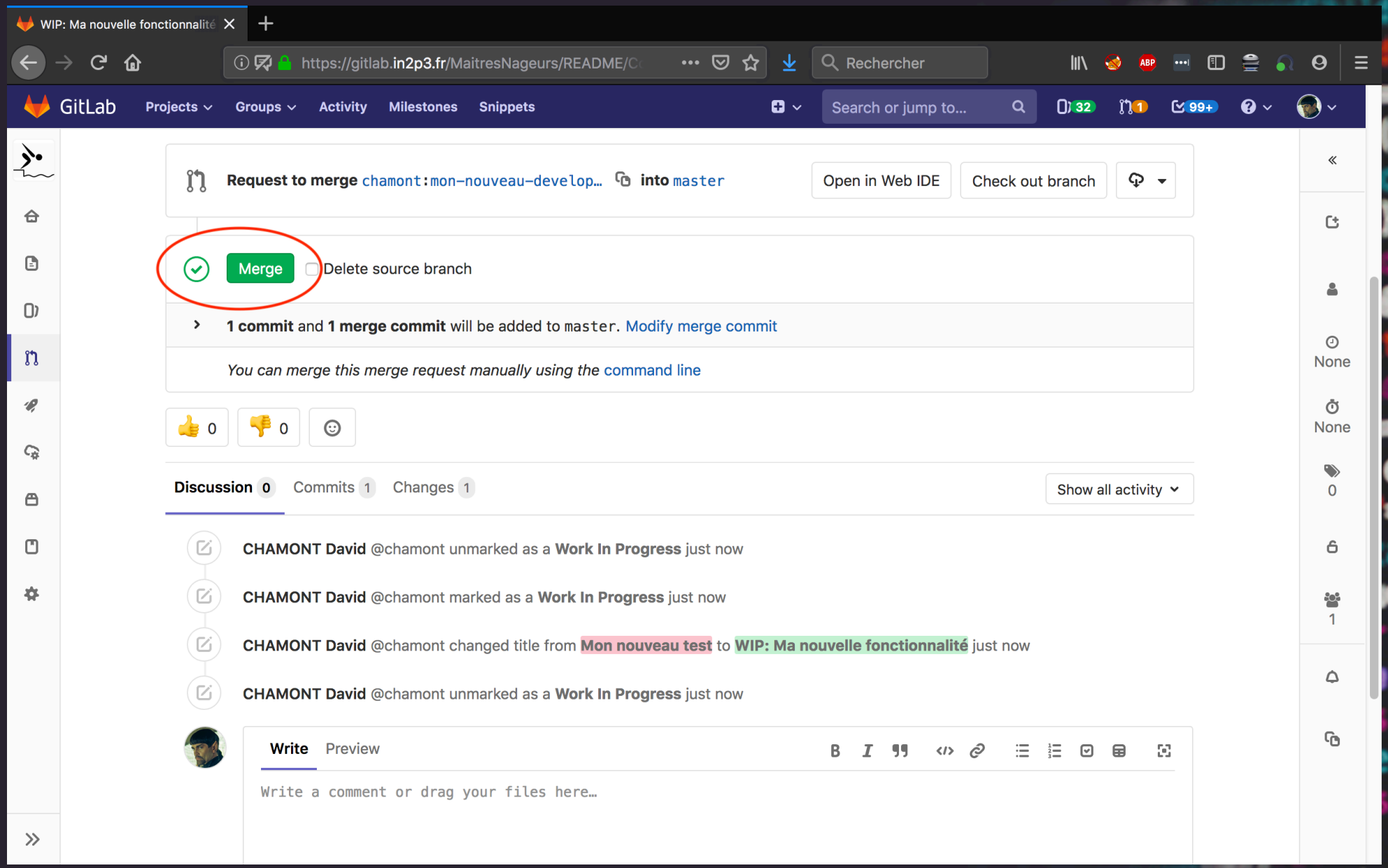

New Merge Request · CHAMON
https://gitlab.in2p3.fr/chamont/CodingPool/merge
Rechercher
GitLab
Projects
Groups
Activity
Milestones
Snippets
Search or jump to...
32
99+
CHAMONT David › CodingPool › Merge Requests › New
New Merge Request
Source branch
chamont/CodingPool
mon-nouveau-developpement
Mon nouveau test
CHAMONT David authored 2 minutes ago
09534de2
Target branch
MaitresNageurs/README/CodingP...
master
Merge branch 'nettoyage_@route' into 'master'
CHAMONT David authored 2 months ago
36e8c4c7
Compare branches and continue


New Merge Request · CHAMON
https://gitlab.in2p3.fr/chamont/CodingPool/merge
Rechercher
GitLab
Projects
Groups
Activity
Milestones
Snippets
Search or jump to...
32
99+
CHAMONT David › CodingPool › Merge Requests › New
New Merge Request
From chamont/CodingPool:mon-nouveau-developpement into MaitresNageurs/README/CodingPool:master
Change branches
Title
WIP: Mon nouveau test
Remove the WIP: prefix from the title to allow this Work In Progress merge request to be merged when it's ready.
Add description templates to help your contributors communicate effectively!
Description
Write
Preview
Write a comment or drag your files here…
Markdown and quick actions are supported
Attach a file
Assignee
Unassigned
Assign to me
Milestone
Milestone


EXPLORER
OPEN EDITORS 1 UNSAVED
Dockerfile CodeGuards/docker
mon_nouveau_test.py CodingP...
test-flask-mail.py CodeGuards/...
config.py CodeGuards/codeguar...
flask_mail.py CodeGuards/venv/...
README.md CodeGuards
CODINGPOOL
.idea
CodeGuards
CodingPool
app
__pycache__
.vscode
static
templates
.gitignore
add_tutos.py
bdd_for_labels.db
codingPool.py
createBdd.sql
get_user_infos.py
labels
mon_nouveau_test.py
my_datas
register.py
sql_request.py
update_labels.py
assets
docker
README.md
CodingPool.wiki
LabelsTower
OUTLINE
Dockerfile
mon_nouveau_test.py
test-flask-mail.py
config.py
flask_mail.py
README.md
global session_db
cipher_suite = Fernet(app.crypt_key)
user_mail = cipher_suite.decrypt(mail_address.encode()).decode("utf-8")
user_mail = user_mail.replace("mail:", '')
user = session_db.query(User).filter_by(mail=user_mail).first()
user.activated = 1
session_db.add(user)
session_db.commit()
return render_template("activate_account.html")
@app.route('/admin')
def admin():
global session_db
user = flask_login.current_user
if (user.is_authenticated):
if (user.mail in admin_list):
return render_template("admin.html")
else:
return ("Vous n'êtes pas administrateur.")
return ("Vous devez être connecté avec un compte administrateur.")
@app.route('/logout')
def logout():
global session_db
user = flask_login.current_user
user.authenticated = 0
session_db.add(user)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: bash
mbp-chamont:app chamont$ ...
master
Python 3.7.2 64-bit
0 0
Ln 260, Col 1
Spaces: 2
UTF-8
LF
Python


EXPLORER
OPEN EDITORS
Dockerfile CodeGuards/docker
mon_nouveau_test.py Cod... M
test-flask-mail.py CodeGuards/...
config.py CodeGuards/codeguar...
flask_mail.py CodeGuards/venv/...
README.md CodeGuards
CODINGPOOL
.idea
CodeGuards
CodingPool
app
__pycache__
.vscode
static
templates
.gitignore
add_tutos.py
bdd_for_labels.db
codingPool.py
createBdd.sql
get_user_infos.py
labels
mon_nouveau_test.py M
my_datas
register.py
sql_request.py
update_labels.py
assets
docker
.gitignore
README.md
test
OUTLINE
Dockerfile
mon_nouveau_test.py
test-flask-mail.py
config.py
flask_mail.py
README.md
218 global session_db
219 cipher_suite = Fernet(app.crypt_key)
220 user_mail = cipher_suite.decrypt(mail_address.encode()).decode("utf-8")
221 user_mail = user_mail.replace("mail:", '')
222 user = session_db.query(User).filter_by(mail=user_mail).first()
223 user.activated = 1
224 session_db.add(user)
225 session_db.commit()
226 return render_template("activate_account.html")
227
228 @app.route('/admin')
229 def admin():
230 global session_db
231 user = flask_login.current_user
232 if (user.is_authenticated):
233 if (user.mail in admin_list):
234 return render_template("admin.html")
235 else:
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: bash
mbp-chamont:CodingPool chamont$ git status
On branch mon-nouveau-developpement
Your branch is up-to-date with 'origin/mon-nouveau-developpement'.
nothing to commit, working directory clean
mbp-chamont:CodingPool chamont$ git pull upstream master
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 10 (delta 3), reused 0 (delta 0)
Unpacking objects: 100% (16/16), done.
From gitlab.in2p3.fr:MaitresNageurs/README/CodingPool
* branch master -> FETCH_HEAD
36e8c4c..1e55e8a master -> upstream/master
Merge made by the 'recursive' strategy.
.gitignore | 2 ++
README.md | 8 ++++++++
test | 0
3 files changed, 10 insertions(+)
create mode 100644 .gitignore
create mode 100644 test
mbp-chamont:CodingPool chamont$
mon-nouveau-developpement* Python 3.7.2 64-bit 0 0 Ln 260, Col 1 Spaces: 4 UTF-8 LF Python


WIP: Ma nouvelle fonctionnalité
https://gitlab.in2p3.fr/MaitresNageurs/README/C
Rechercher
GitLab
Projects
Groups
Activity
Milestones
Snippets
Search or jump to...
MaitresNageurs › README › CodingPool › Merge Requests › !84
Open
Opened 2 minutes ago by CHAMONT David
Edit
Close merge request
WIP: Ma nouvelle fonctionnalité
Pour la démo...
Edited just now by CHAMONT David
Request to merge chamont:mon-nouveau-develop… into master
Open in Web IDE
Check out branch
Merge
This is a Work in Progress
Resolve WIP status
You can merge this merge request manually using the command line
0
0
Discussion 0
Commits 1
Changes 1
Show all activity
CHAMONT David @chamont unmarked as a Work In Progress just now
CHAMONT David @chamont marked as a Work In Progress just now
CHAMONT David @chamont changed title from Mon nouveau test to WIP: Ma nouvelle fonctionnalité just now
None
None
0
1


WIP: Ma nouvelle fonctionnalité
https://gitlab.in2p3.fr/MaitresNageurs/README/C
Rechercher
GitLab
Projects
Groups
Activity
Milestones
Snippets
Search or jump to...
Request to merge chamont:mon-nouveau-develop… into master
Open in Web IDE
Check out branch
Merge
Delete source branch
1 commit and 1 merge commit will be added to master. Modify merge commit
You can merge this merge request manually using the command line
Discussion 0
Commits 1
Changes 1
Show all activity
CHAMONT David @chamont unmarked as a Work In Progress just now
CHAMONT David @chamont marked as a Work In Progress just now
CHAMONT David @chamont changed title from Mon nouveau test to WIP: Ma nouvelle fonctionnalité just now
CHAMONT David @chamont unmarked as a Work In Progress just now
Write
Preview
Write a comment or drag your files here…

A quoi bon ? (j'analyse seul)

- Pour archiver vos versions successives et pouvoir revenir en arrière.
- Pour pouvoir mener plusieurs développements indépendants dans plusieurs branches.
- Pour avoir des tests exécutés automatiquement en continu et détectés un problème dès qu'il arrive.
- Pour gérer mes tâches à l'aide du panneau de tickets.
- Pour avoir une image prêt à distribuer, et une analyse reproductible...

Pour aller plus loin

à l'IN2P3 et autour

Ecole de GIF, septembre 2019
David Chamont, LAL

Ressources de calcul (hors-grille)

- Pour de la R&D :
  - Plateforme P2IO-ACP (GPUs, FPGAs, Intel/AMD, …),
  - Techlab du CERN, ...
- Pour de la Production au CC (chiffres 2018)
  - GPGPU cluster : Infiniband, 10 C4130 K80 => 40 GPUs
  - HPC cluster : Infiniband, 16 x C6320 => 512 coeurs
- Supercalculateurs :
  - centres TGCC, IDRIS, CINES,
  - programme européen PRACE.
- FG-Cloud (au sein de France-Grille)

Veille technologique

- Conférences
  - CHEP : http://chep2019.org/
  - ACAT : https://indico.cern.ch/event/708041/
  - JCAD : https://jcad2019.sciencesconf.org/
- Réseaux métiers
  - Groupe Calcul : https://calcul.math.cnrs.fr/
  - Devlog : http://devlog.cnrs.fr/

R&D informatique IN2P3

- Decalog / ComputeOps : machines virtuelles, cloud, conteneurs (Docker, Singularity, …).
- Decalog / Reprises : nouveaux matériels de calcul, portabilité des codes et reproductibilité numérique.
- CompStat : apprentissage automatique.
- *AstroLab Software : données massives.*
- Groupe de discussion autour du Calcul Quantique
- Participations à Aida2020, Asterics, Mastodons...

Prospectives Nationales IN2P3 : https://prospectives2020.in2p3.fr/

HEP Software Fundation

- Le Community White Paper de 2017
  https://link.springer.com/article/10.1007%2Fs41781-018-0018-8
- Groupes de travail : Data Analysis, Detector Simulation, Frameworks, Physics Generators, Packaging, Python in HEP, Quantum Computing ,Reconstruction and Software Trigger , Software/Developer Tools, Training, Visualization.
  https://hepsoftwarefoundation.org/what_are_WGs.html
- Journal Computing and Software for Big Science
  https://link.springer.com/journal/41781

Formation permanente

- Groupe Calcul : https://formation-calcul.fr/
- ANFs, dont l'Ecole Informatique IN2P3, annuelle
  - 2018 : les conteneurs
  - 2019 : programmation fonctionnelle
    http://formation.in2p3.fr/Info19/ProgFonct19.html
- Tutoriels en ligne, à l'IN2P3 et ailleurs
  - https://gitlab.in2p3.fr/MaitresNageurs
  - *https://codingpool.in2p3.fr/*
- Formation permanente CNRS : Git, C++, Python...
  - https://formation.ifsem.cnrs.fr/

Pour aller moins loin

comment coder aujourd'hui ?

Ecole de GIF, septembre 2019
David Chamont, LAL

Au quotidien, sur votre poste

- Programmez simple et fonctionnel,
  organisez vos données en SoA,
  cela sera plus facile à paralléliser.
- Exécutez dans un conteneur,
  cela sera plus facile à déployer.
- Soignez la reproductibilité,
  cela sera plus facile à vendre.

Besoin de performance ?

- Votre problème est-il IO-Bound ?
- Sinon, n'optimisez pas en aveugle. Profilez !

Gare à l'arithmétique flottante !

- En Python, 0.1+0.2 => 0.30000000000000004
- Cours un peu ancien : https://hal.inria.fr/inria-00071477
- Ecole thématique PRECIS :
  - 2013 : https://calcul.math.cnrs.fr/2013-03-ecole-precis.html
  - 2017 : https://precis.sciencesconf.org/
- Des outils d'arithmétique stochastique pour détecter vos erreurs :
  - VERROU
  - CADNA
  - VERIFICARLO