

Jenkins

Etape 1 : Push du Jenkinsfile sur l'arborescence du projet

```
Jenkinsfile
1  pipeline{
2      agent any
3      stages{
4          // Install python packages *
5          stage('Python Requirements'){
6              steps{
7                  sh 'python3 -m venv env'
8                  sh '. env/Scripts/activate'
9                  sh 'pip install -r requirements.txt'
10             }
11         }
12         // Run the tests
13         stage('Run Pytest'){
14             steps{
15                 sh '. env/Scripts/activate'
16                 sh 'cd env && python3 -m tests.test_calculator'
17             }
18         }
19     }
20 }
```

Etape 2 : Mise en place d'une image et d'un container Docker

Dockerfile :

```
FROM jenkins/jenkins:lts-jdk11
USER root
RUN apt-get update && apt-get upgrade -y
# Install PIP and Python
RUN apt-get install -y python3-pip
# Upgrade PIP to the latest version
RUN pip install --upgrade pip
```

Build the image :

`docker build -t jenkins-image .`

Démarrer le container :

`docker run --name jenkins-python -d -v jenkins_home:/var/jenkins_home -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins-image`

Obtenir la clé Jenkins pour la configuration :

`docker exec jenkins-python cat /var/jenkins_home/secrets/initialAdminPassword`

```
[louis@pavillon Python Calculatrice]$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
de40f2d9dde4   maildev/maildev "bin/maildev"           14 minutes ago Up 14 minutes (healthy) 0.0.0.0:1025->1025/tcp, :::1025->1025/tcp, 0.0.0.0:1080->1080/tcp, :::1080->1080/tcp hungry_hugle
9e132caf29ac   jenkins-image  "/usr/bin/tini -- /u-"  6 days ago    Up 14 minutes  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp jenkins-python
[louis@pavillon Python Calculatrice]$
```

Etape 3 : Configuration de Jenkins

Création d'un nouveau item, puis configuration de la pipeline

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/ys08/b3-c2-dev-tu-alary-guillaume

Credentials ?

- none -

Add

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

main

Add Branch

Etape 3 : Mise en place de notifications sur Discord en cas d'échec de test

Nous commençons par mettre en place un test qui échoue volontairement

```
# Fail as the result is incorrect
def test_squareroot_in_substraction_error(self):
    self.assertEqual(Calculator.calculate("2-√4"), 2)
```

Nous allons ensuite configurer Jenkins pour envoyer un mail lorsque les tests échouent.

- Vérifier que le plugin *Email-ext* est bien activé dans les paramètres Jenkins.

Configuration de la notification par mail dans les paramètres de Jenkins

E-mail Notification

SMTP server

smtp-relay.gmail.com

Default user e-mail suffix ?

Advanced ^

Edited

☒ Use SMTP Authentication ?

User Name

louis.alary

Password

 Concealed

☒ Use SSL ?

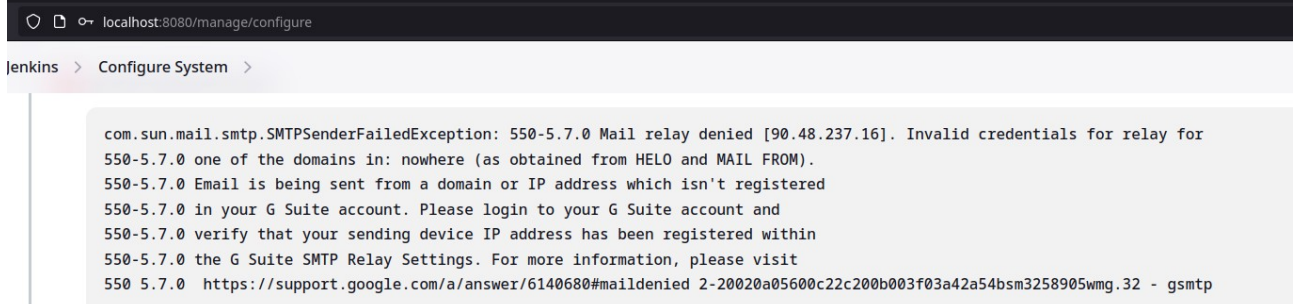
☐ Use TLS

SMTP Port ?

465

Après avoir tenté de faire fonctionner l'envoi d'email avec MailDev, Outlook, Gmail et d'avoir des problèmes avec les 3 services :

- Envoi d'email désactivé sur le compte Outlook de l'EPSI



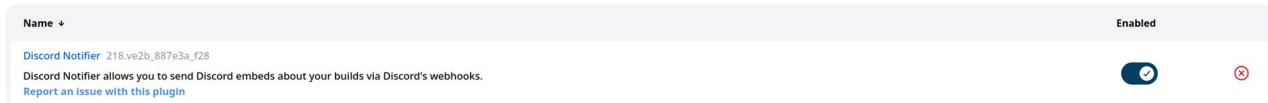
- Besoin d'un compte administrateur avec Gmail

- MailDev pas fonctionnel

Voici le Jenkinsfile censé fonctionner

```
Jenkinsfile
1 pipeline{
2   agent any
3   stages{
4     // Install python packages *
5     stage('Python Requirements'){
6       steps{
7         sh 'python3 -m venv env'
8         sh '. env/Scripts/activate'
9         sh 'pip install -r requirements.txt'
10      }
11    }
12    // Run the tests
13    stage('Run Pytest'){
14      steps{
15        sh '. env/Scripts/activate'
16        sh 'cd env && python3 -m tests.test_calculator'
17      }
18    }
19  }
20  post {
21    failure {
22      mail to: 'louis.alary@epsi.fr',
23           subject: "Failed Pipeline: ${currentBuild.fullDisplayName}",
24           body: "Something is wrong with ${env.BUILD_URL}"
25    }
26  }
27 }
```

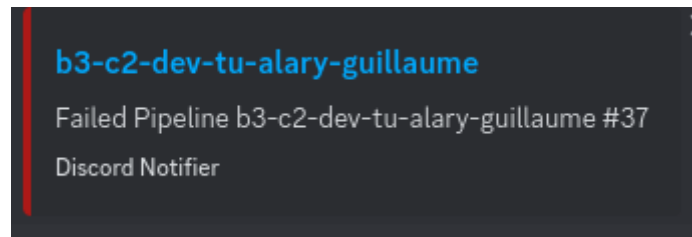
Nous avons décidé d'utiliser les notifications avec Discord.
Pour se faire, nous installons d'abord le plugin "Discord Notifier"



Nous reconfigurons le Jenkinsfile pour utiliser Discord Notifier

```
Jenkinsfile
1 pipeline{
2   agent any
3   stages{
4     // Install python packages *
5     stage('Python Requirements'){
6       steps{
7         sh 'python3 -m venv env'
8         sh '. env/Scripts/activate'
9         sh 'pip install -r requirements.txt'
10      }
11    }
12    // Run the tests
13    stage('Run Pytest'){
14      steps{
15        sh '. env/Scripts/activate'
16        sh 'cd env && python3 -m tests.test_calculator'
17      }
18    }
19  }
20  post {
21    failure {
22      discordSend description: "Failed Pipeline ${currentBuild.fullDisplayName}",
23                  link: env.BUILD_URL,
24                  footer: "Discord Notifier",
25                  title: env.JOB_NAME,
26                  webhookURL : "https://discord.com/api/webhooks/1092712641128042576/R4DeY0BLeIgjgEATd"
27    }
28  }
29 }
30
31
32 }
```

Nous lançons la pipeline et nous obtenons le message d'erreur sur Discord



Etape 4 : Lancement des tests à chaque push de code

Pour lancer les tests à chaque push de code, il faut créer un webhook sur Github qui fait référence à Jenkins.

Nous utilisons Jenkins dans un Docker. Il faut donc exposer l'adresse IP et le port du container pour qu'il fonctionne avec Github.

Malheureusement, nous n'avons pas réussi à exposer l'adresse du container, les tests doivent être lancés manuellement par l'utilisateur.

Voici quand même le schéma d'intégration dans le cas où le webhook sur Github fonctionnerait.

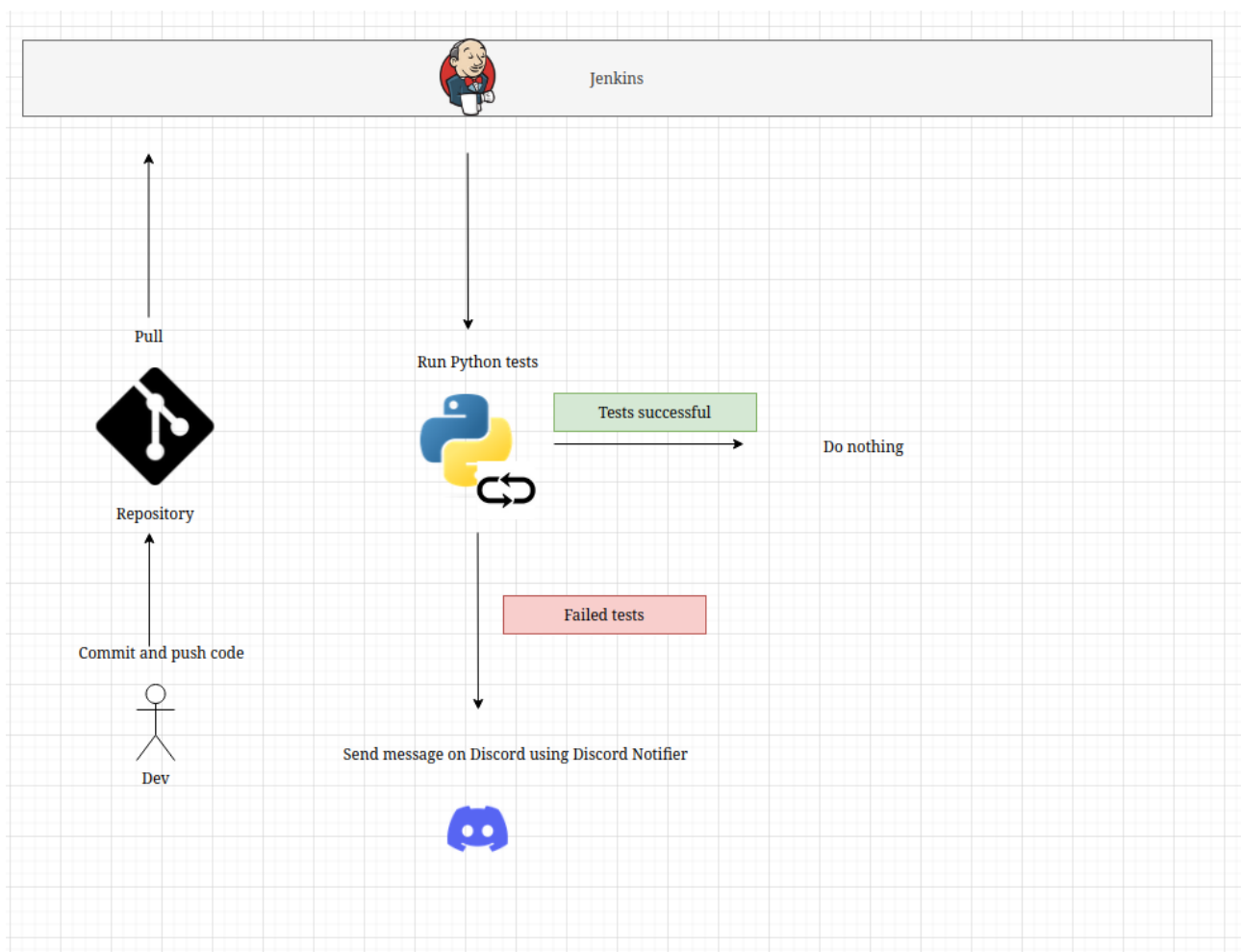


Schéma intégration continue