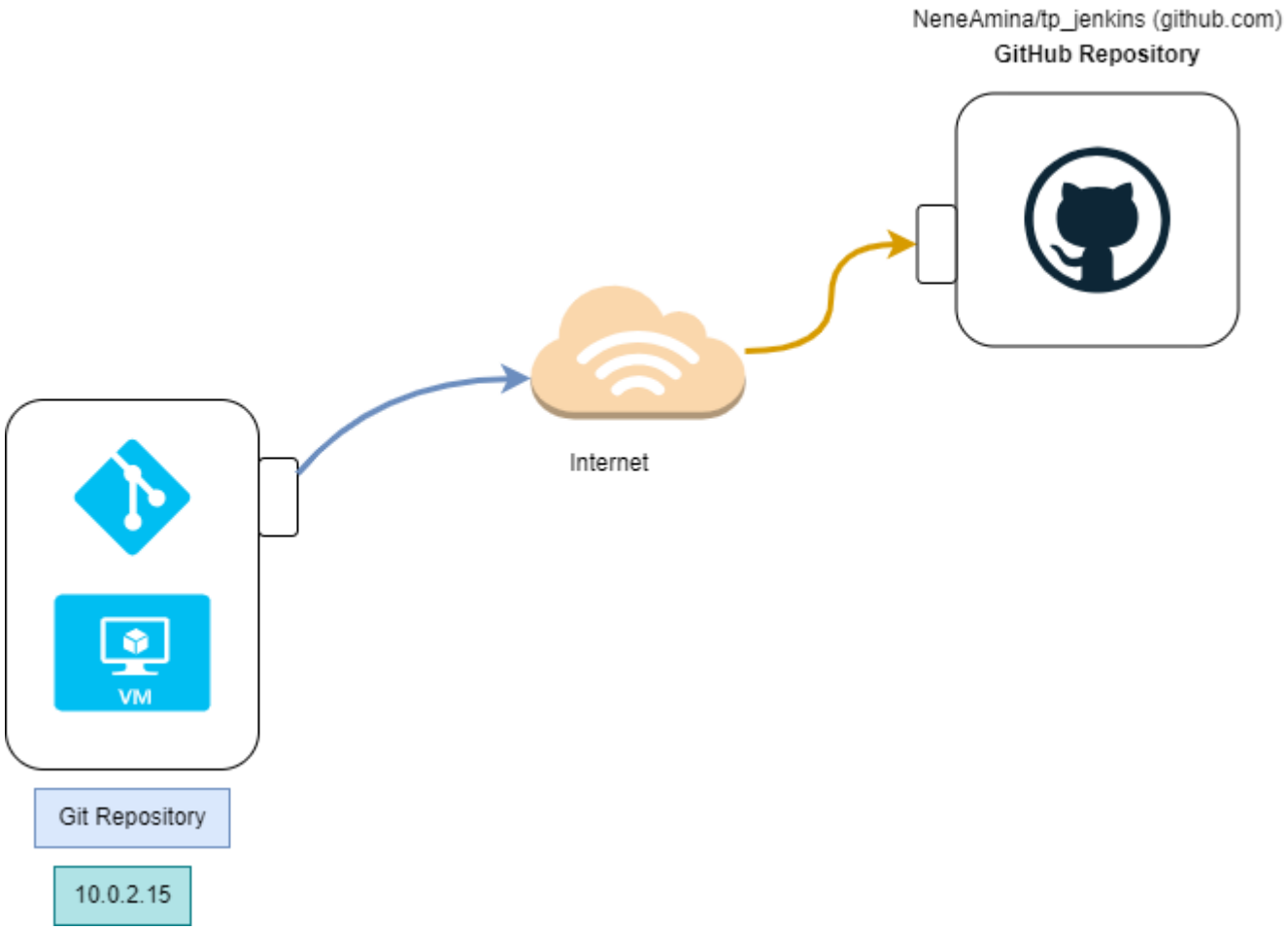
Nene Aminatou Diallo

Rimey Aboky

Docker jenkins

1. Architecture mise en place repertoire github



1.1 Mise en place du repertoire de travail

Nous allons migrer le repertoire de test vers un repertoire dont nous avons le contraire. nous clone le repertoire de test

```
git clone https://github.com/karimsahebettabaa/jenkins-docker
```

Ensuite nous pointons l'url du repo de test vers le nouveau repo que nous avons créé repertoire de travail.

```
git remote set-url origin https://github.com/NeneAmina/tp_jenkins
```

Maintenant que le repertoire de travail pointe sur notre propre repo github, nous passons à l'installation docker

1.2 Installation docker

Nous effectuons l'installation en suivant la documentation officielle sur le site officiel docker installation

1.3 Conteniriser l'applicaition

A partir du dockerfile, nous créons une image du container que nous allons deployé.

```
docker build -t myapp/flask .
```

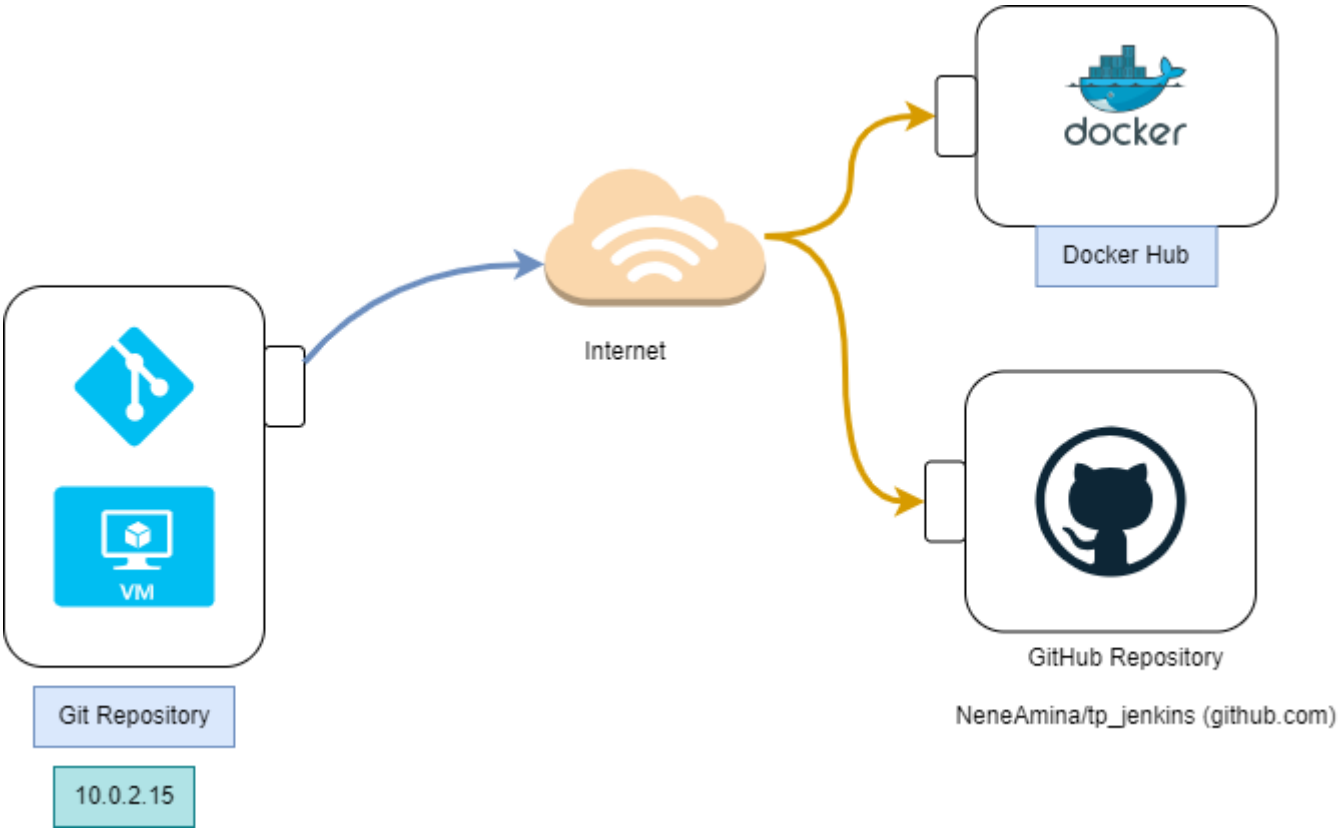
Ensuite nous demarrons le conteneur

```
docker run -d -p 8081:8081 myapp/flask
```

l'application tourne sur localhost:8081 accessible sur http://localhost:8081

maintenant que l'application tourne sur un conteneur, nous allons créér un registre de gestion d'image sur docker hub en suivant la documentaition officiel https://hub.docker.com/

2 Architecture du setup avec docker hub

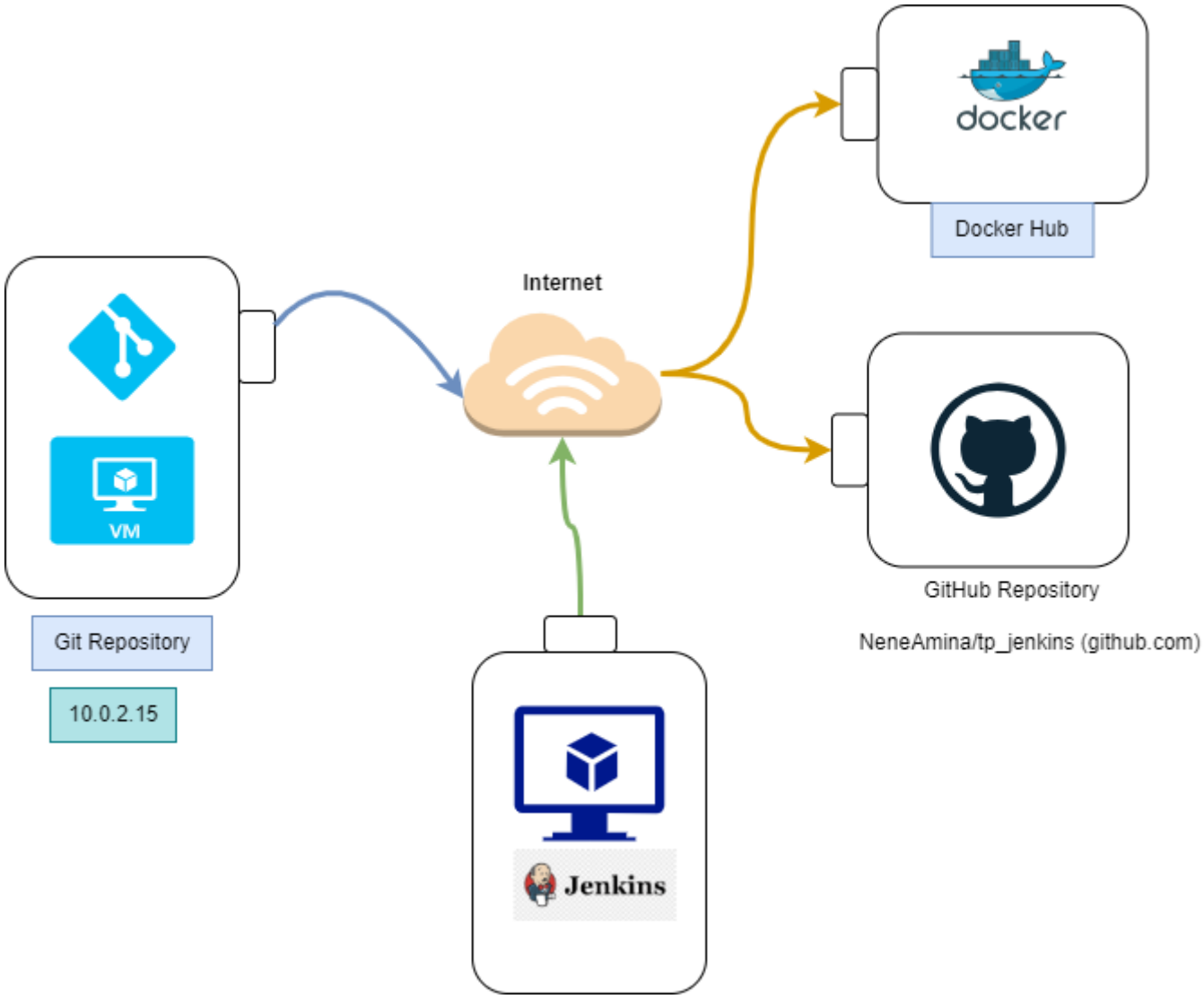


Maintenant que nous avons notre repertoire github en place et notre registre d'image sur docker hub nous allons mettre à jour notre fichier de build jenkinsfile. Nous remplaçons dans le fichier la variable

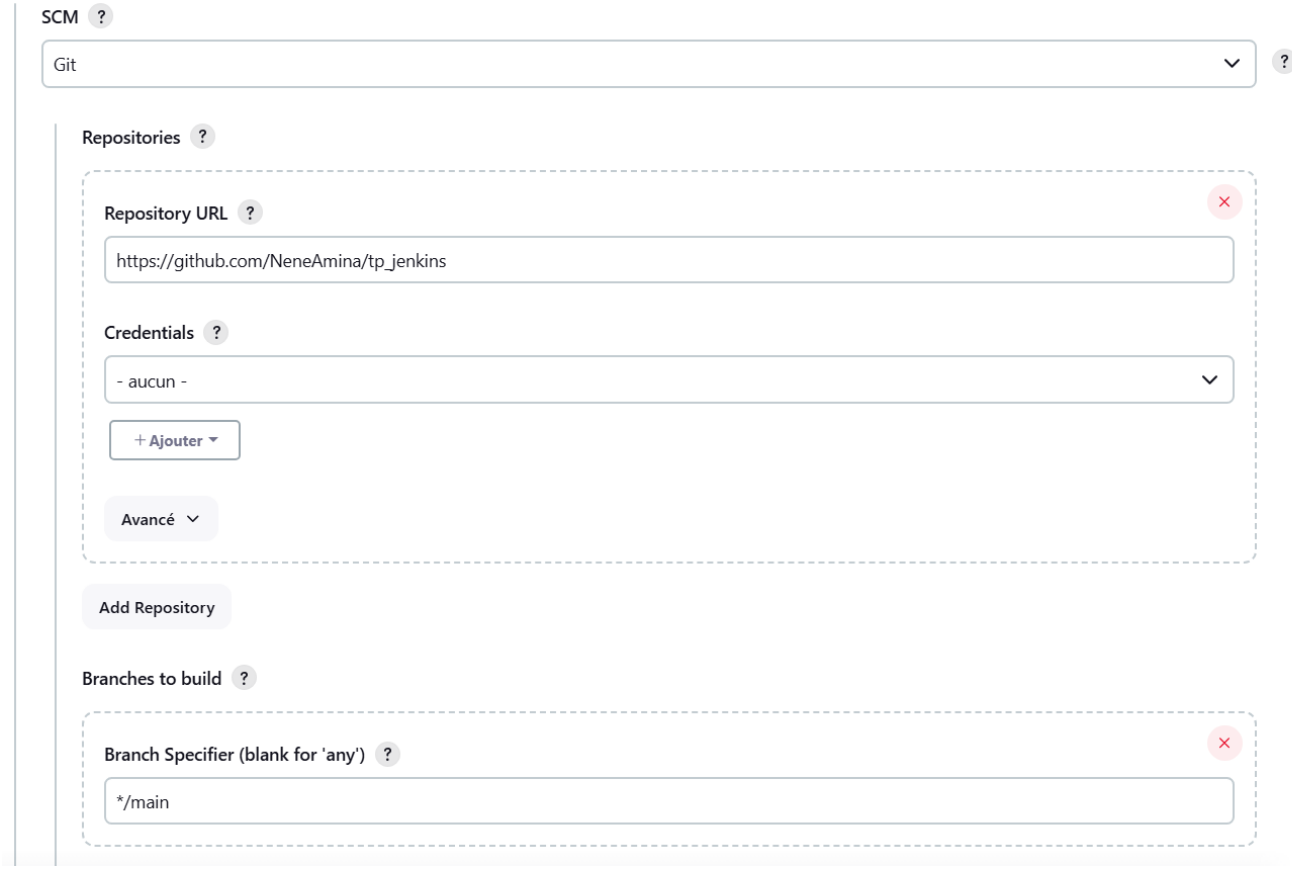
{REGISTER_DOCKER_HUB} par le nom de registre docker hub.

```
pipeline {
    agent any

    stages {

        stage('Build docker image') {
            steps {
                sh 'docker build -t {REGISTRE_DOCKER_HUB}:$BUILD_NUMBER .'
            }
        }

        stage('Login to DockerHub') {
            steps {
                sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u
$DOCKERHUB_CREDENTIALS_USR --password-stdin'
            }
        }

        stage('Push image') {
            steps {
                sh 'docker push {REGISTRE_DOCKER_HUB}:$BUILD_NUMBER'
            }
        }
    }

    post {
        always {
            sh 'docker logout'
        }
    }
}
```

3 Architecture avec jenkins integré



3.1 Configuration jenkins



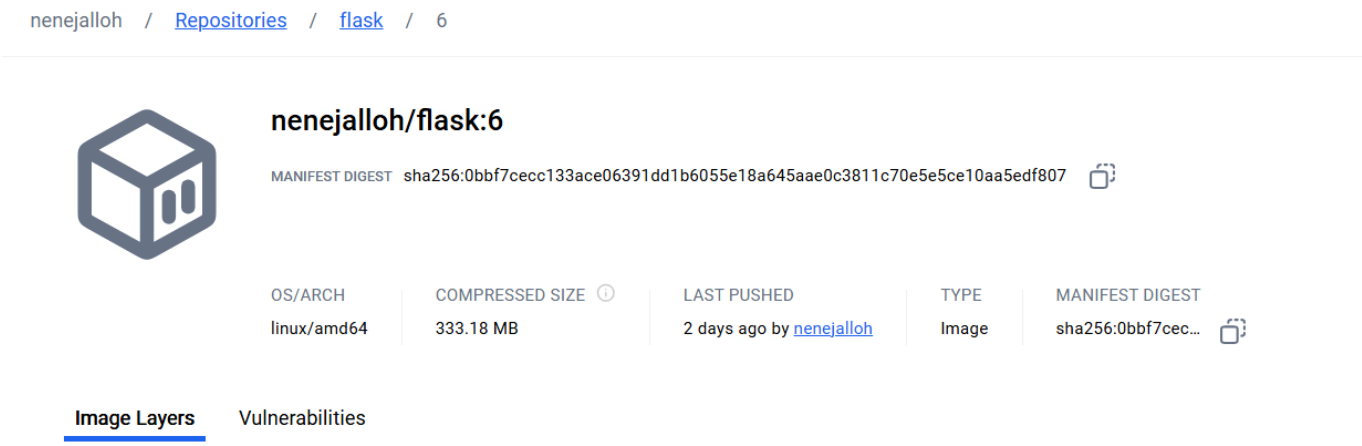
Ensuite nous l'execution du pipeline après avoir configuré les variables d'environnement dans notre système jenkins.

3.2 Différents étapes du build

Step	Arguments	Status
Start of Pipeline - (32 s in block)		
node - (25 s in block)		
node block - (23 s in block)		
stage - (2,3 s in block)	Declarative: Checkout SCM	
stage block (Declarative: Checkout SCM) - (1,8 s in block)		
checkout - (1,5 s in self)		
withEnv - (21 s in block)	GIT_BRANCH, GIT_COMMIT, GIT_PREVIOUS_COMMIT, GIT_URL	
withEnv block - (21 s in block)		
stage - (8,6 s in block)	Build docker image	
stage block (Build docker image) - (8,3 s in block)		
sh - (7,8 s in self)	docker build -t nenejalloh/flask:$BUILD_NUMBER .	
stage - (2,1 s in block)	login to dockerhub	
stage block (login to dockerhub) - (1,9 s in block)		
stage - (8,6 s in block)	Build docker image	
stage block (Build docker image) - (8,3 s in block)		
sh - (7,8 s in self)	docker build -t nenejalloh/flask:$BUILD_NUMBER .	
stage - (2,1 s in block)	login to dockerhub	
stage block (login to dockerhub) - (1,9 s in block)		
sh - (1,7 s in self)	echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin	
stage - (8,5 s in block)	push image	
stage block (push image) - (8 s in block)		
sh - (7,5 s in self)	docker push nenejalloh/flask:$BUILD_NUMBER	
stage - (1 s in block)	Declarative: Post Actions	
stage block (Declarative: Post Actions) - (0,87 s in block)		
sh - (0,76 s in self)	docker logout	

3.4 Résultat build output

3.5 Vérification sur dockerhub



4 Amelioration

Pour faciliter le déploiement aujourd'hui github dispose directement d'une solution de CI/CD nommé github actions avec laquel nous avons pas à installer et configurer un système intermédiare comme jenkins.L'avantage est que nous avons un système de moins à gérer et pas de plugin spécifique à recherche. Lien vers github actions https://docs.github.com/fr/actions