

MÉTHODOLOGIE DEVOPS. AMÉLIORER LA PRODUCTIVITÉ D'UNE INFRASTRUCTURE EXISTANTE POUR LE SAAS

Sommaire

- Problématique de l'entreprise
- Infrastructure Actuelle (Schéma)
- Limites de l'infrastructure
- Indicateurs de performances du projet.
- Nouvelle Infrastructure AWS (Schéma)
- Tâches du projet, RACI & GANTT
- Tableau Budgétaire (Pour la direction uniquement)
- Risques encourus
- Conclusion

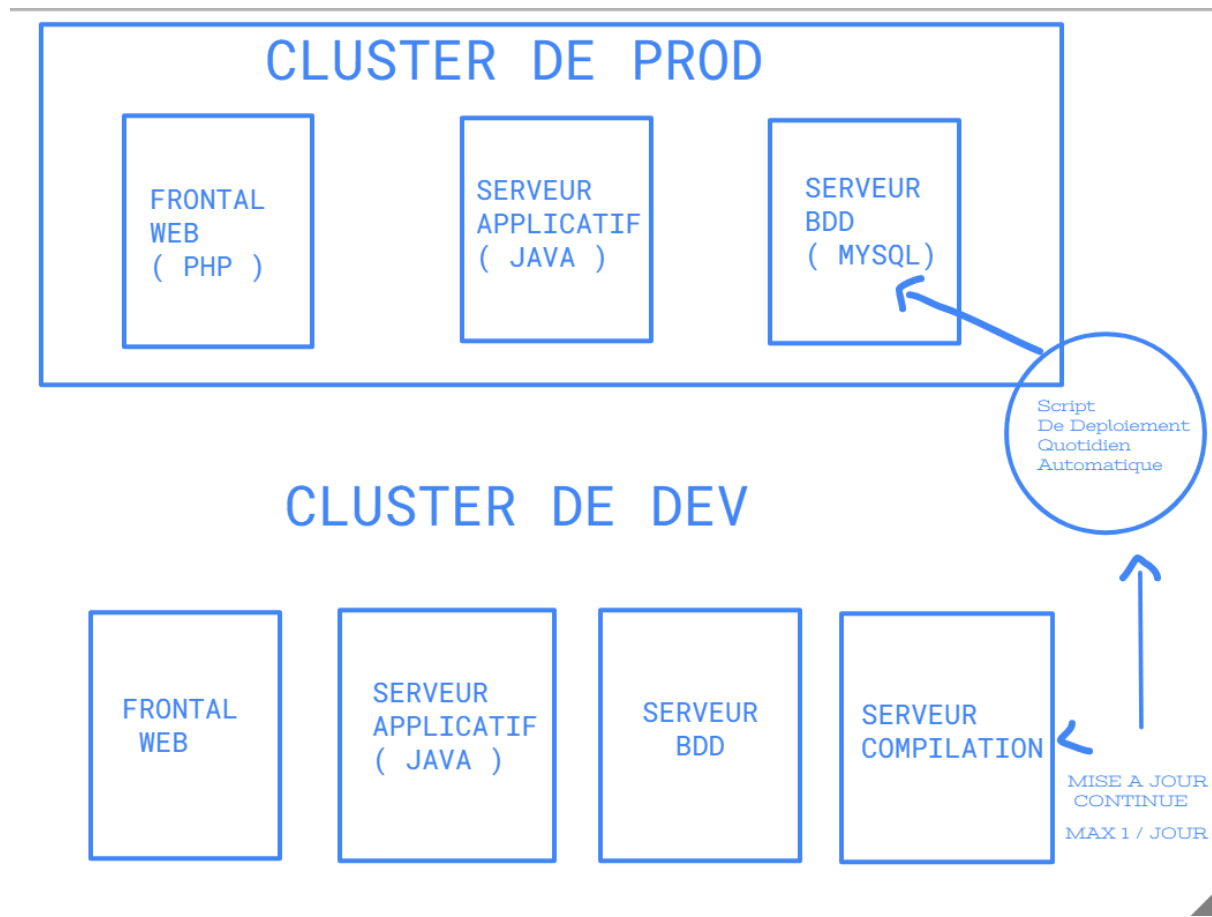
Problématique de l'entreprise :

L'entreprise édite un logiciel SaaS et souhaiterait s'orienter vers une démarche DevOps afin de gagner en agilité et raccourcir son "Time to Market". Actuellement, la compilation du logiciel prend 6 heures et se fait de nuit sur les serveurs des outils internes quand ces derniers ne sont pas utilisés.

Dans le cadre d'une orientation DevOps, le but serait de pouvoir compiler le logiciel plusieurs fois par jour sans impacter le serveur des outils internes en terme de charge machine et à un coût modéré.

L'entreprise décide de lancer un projet pilote sur une plate-forme IaaS sur le cloud afin de bénéficier de l'élasticité offerte par cette plate-forme.

Infrastructure Actuelle du SAAS :

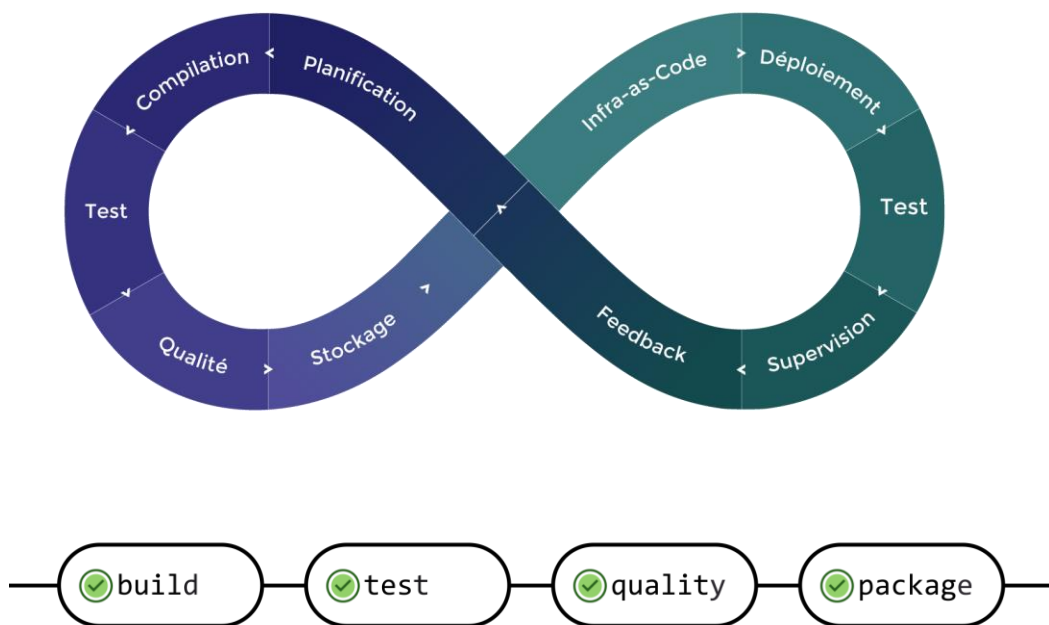


Limites de l'infrastructure actuelle :

- Pas d'intégration & tests en continu
- Monopolisation du serveur lors de la compilation
- Limite du nombre de compilation par jour.
- Pas d'élasticité des serveurs de dev ou prod en cas de grosse mises à jour ou de grosse charge de trafic.
- Aucune supervision et remonté d'alerte de L'APP afin de garantir un cycle continu.

Indicateurs de performances :

#1 Avantages IAAS AWS :



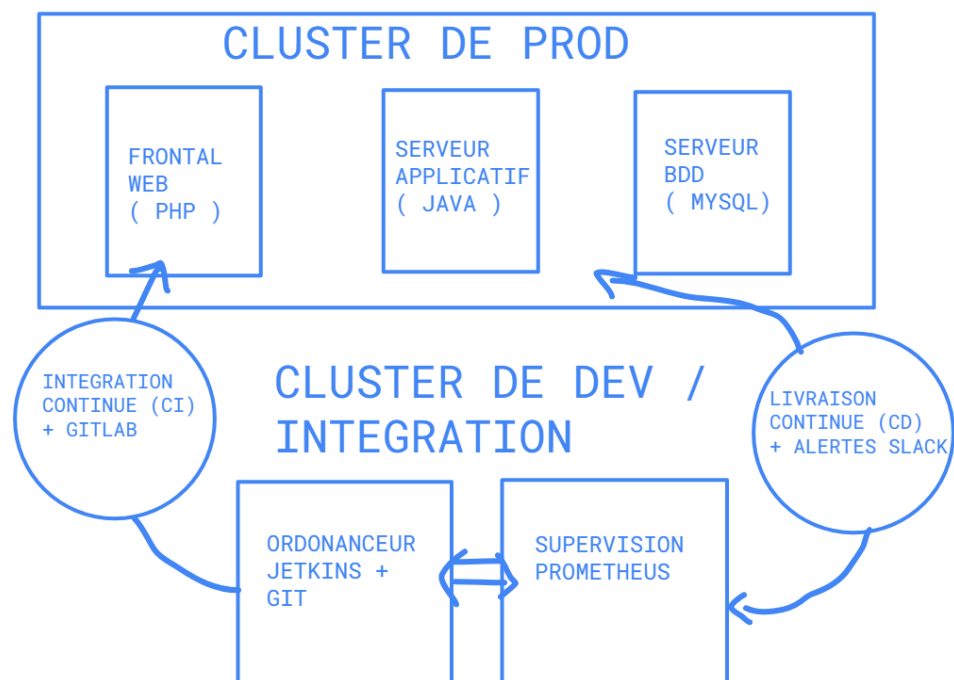
Problèmes des points ci-dessus résolus

- Qualité / Robustesse Code
- Temps de compilation
- Nombre de compilation par jour
- Suivi du développement

#2 Bon déroulement du projet :

- Respect de la deadline
- Respect des attribution des tâches
- Respect du cahier des charges pour la réalisation techniques
- Respect des coûts
- Respect de la continuité de service ou interruption dans des moments à faible trafic.

Infrastructure futur du SAAS sur Cloud AWS :



Tâches du projet :

ETAPE 1 : IDENTIFIER LES ADHÉRENCES DU SERVEUR APPLICATIF

Trouver les flux entre les machines des développeurs & le serveur applicatif actuel

ETAPE 2 : SCHÉMATISER L'IAAS CLOUD :

3 SERVEURS DE PROD (FRONT + SERV APP + BDD) + 2 SERVEURS DE DEV (ORDONNANCEUR / GIT [GITLAB + SLACK] & SERVEUR DE SUPERVISION [PROMETHEUS])

ETAPE 3 : CRÉATION DU LAN IAAS AWS :

Définir la topologie du réseau et créer les différents VPC AWS (Réseau, Sous Réseau, IP élastique) Apporter également la sécurité entre les serveurs grâce au firewall AWS.

ETAPE 4 : CRÉATION DES SERVEURS DANS GROUPE AUTOSCALING :

Créer les images des différents serveurs et intégrer ces derniers dans un groupe de mise à l'échelle afin de lancer X serveurs identiques dans le cas de mises à jour importantes ou transverses.

ETAPE 5 : INSTALLATION DES PAQUETS & DÉPENDANCES PROPRES AU RÔLE DE CHAQUE SERVEUR :

Mettre à jour les serveurs, optimiser ces derniers pour ne garder que les services nécessaires et précédemment installés selon le principe de minimisation.

ETAPE 6 : CONFIGURATION DES SERVEURS POUR ASSURER UN SAAS AUTOMATISÉ, CONTINUE & SUPERVISÉ

Configurer Apache, Mysql/Nginx & importer la base de donnée à jour que l'on pourra garder en copie sur un bucket S3. Vérifier les flux entrant & sortant du serveur applicatif tomcat par analyse des trames réseaux. Déployer l'ordonnanceur Jenkins, Dovecot, Git & GitLab et intégrer Slack à GitLab.

ETAPE 7 : TEST DE FONCTIONNEMENTS & MESURE DES PERFORMANCES SUR INFRA CLOUD & ANCIENNE INFRA :

D'après la méthodologie C.A.L.M.S il est primordiale de mesurer l'ensemble des taches du SI pour améliorer la productivité du SAAS. Il faudra donc faire des compilations, déployer les mises à jour sur l'ancienne Infrastructure & répéter le processus sur la nouvelle infra pour constater le delta des performances. Une matrice pourra être créer afin de donner une estimation en % des performances mesurés ainsi que la vélocité à laquelle une mise à jour pourra être déployé. On profitera de ses test de performances pour vérifier que l'infrastructure cloud fonctionne correctement et remonte également les éventuels problèmes comme par exemple une mauvaise qualité du code par des alertes automatisés.

ETAPE 8 : FORMATION DES DEV & OPS A LA NOUVELLE POLITIQUE C.A.L.M.S :

Afin de « Casser » le mur de l'incompréhension entre l'équipe des DEV & L'équipe en charge de l'infrastructure, une réunion avec une hygiène devops est primordiale pour la cohérence du projet sur le long terme. Pour donner les bonnes pratiques et outils cette réunion vise l'ensemble des acteurs du projet.

ETAPE 9 : MISE EN PRODUCTION :

Après avoir validé avec le chef de projet que tout est en ordre, les DNS pourront être redirigés vers le frontal web (IP élastique public pointant sur l'ip privé du frontal web de prod)

ETAPE 10 : BILAN DU PROJET (VALUATION DES INDICATEURS DU PROJET, ANALYSE DES POINTS FORTS ET AMÉLIORATION, COMMUNICATION AUPRÈS DE LA DIRECTION) :

À la suite des différentes mesures, réalisations des taches par les différents acteurs, un document de synthèse devra être rédigé par chacun afin de montrer les forces et les limites de l'infrastructure actuel. Emettre également certaines suggestions d'amélioration après une semaine de production cloud du saas visant à améliorer le cycle continue de développement du projet.

Répartition des tâches à réaliser : (Matrice RACI)

**Chaque tâche donne lieu à la rédaction d'une documentation*

ETAPE	TACHE	CDP	Admin	Dev	Testeur	Superviseur
1	IDENTIFIER LES ADHÉRENCES DU SERVEUR APPLICATIFS	A	R	R	I	I
2	SCHÉMATISER L IAAS CLOUD	A/R	R	C	I	C
3	CRÉATION DU LAN IAAS AWS AVEC TOPOLOGIE RÉSEAU	A	R	I	I	I
4	CRÉATION DES SERVEURS DANS GROUPE AUTOSCALING DANS LAN IAAS AWS	A	R	I	I	I

5	INSTALLATION DES PAQUETS & DÉPENDANCES PROPRE AU RÔLE DE CHAQUE SERVEUR	A	R	R	I	I
6	CONFIGURATION DES SERVEURS POUR ASSURER UN SAAS AUTOMATISÉ, CONTINUE & SUPERVISÉ	A	R	C	I	I
7	TEST DE FONCTIONNEMENTS & MESURE DES PERFORMANCES	A	C	I	R	R
8	FORMATION DES DEV & OPS A LA NOUVELLE POLITIQUE C.A.L.M.S	A/R	I	I	I	R
9	MISE EN PRODUCTION	A	R	C	I	C
10	BILAN DU PROJET	A/R	R	R	R	R

Planification de la migration (Diagramme de Gantt) :

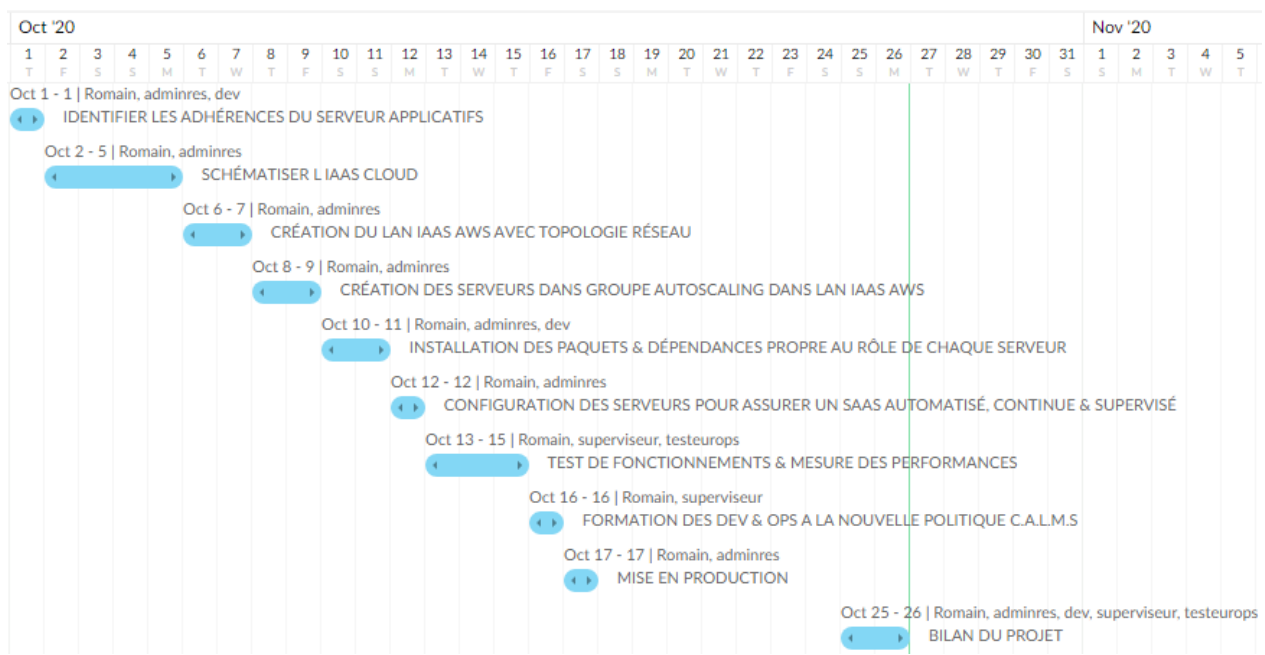


Tableau budgetaire (matrice , jour/homme/etape/coût)

* Chaque journée de travail = 7H

* Lors de la matrice RACI, certains employés sont à disposition des étapes en tant que "consultant".

* Les tâches prennent en compte le temps de rédaction de leurs documentations.

ETAPE	TÂCHE	CDP	ADMIN	DEV	TESTEUR	SUPERVISEUR	Jour/homme
<u>1</u>	IDENTIFIER LES ADHÉRENCES DU SERVEUR APPLICATIFS	0,125	1	1			2,125
<u>2</u>	SCHÉMATISER L IAAS CLOUD	4	4	0,5		0,5	9
<u>3</u>	CRÉATION DU LAN IAAS AWS AVEC TOPOLOGIE RÉSEAU	0,25	2				2,25
<u>4</u>	CRÉATION DES SERVEURS DANS GROUPE AUTOSCALING DANS LAN IAAS AWS	0,25	2				2,25
<u>5</u>	INSTALLATION DES PAQUETS/ DÉPENDANCES PROPRE AU RÔLE DE CHAQUE SERVEUR	0,25	2	2			4,25
<u>6</u>	CONFIGURATION DES SERVEURS POUR ASSURER UN SAAS AUTOMATISÉ, CONTINUE & SUPERVISÉ	0,125	1	0,75			1,875
<u>7</u>	TEST DE FONCTIONNEMENT	0,75	0,75		2	2	5,5

	TS & MESURE DES PERFORMANCES						
<u>8</u>	FORMATION DES DEV & OPS A LA NOUVELLE POLITIQUE C.A.L.M.S	0,5	1	1	1	0,5	4
<u>9</u>	MISE EN PRODUCTION	0,125	1	0,5		0,5	3,125
<u>10</u>	BILAN DU PROJET	1	1	1	1	1	5

** Matrice budgétaire à destination de la direction uniquement. Enlever le coût horaire / poste lors de la réunion de pilotage projet.*

**Cds = Chef de supervision ou Superviseur*

	Cdp	Admin	Dev	Testeur	Cds	Total
Tarif	250	225	160	190	190	
Nbj	7,375	15,75	6,75	4	4,5	38,375
Total	1843,75	3543,75	1080	760	855	8082,5

Risques encourus :

- Respect du temps de projet

--> *Risque Acceptable*

- **Absence / retard d'un ou de plusieurs acteurs**

--> *Risque Acceptable*

- **Problèmes techniques**

--> *Utilisation de solution éprouvée*

- **Performance attendu non présente**

--> *Analyses & mesures des goulots d'étranglements.*

Conclusion :

Grâce à cette nouvelle infrastructure cloud automatisé dans sa mise à l'échelle et dans son système de supervision, l'application sera désormais en mesure de s'industrialiser sans risques majeurs. La qualité et la robustesse du code sera désormais vérifié par une batterie de test unitaires. Plusieurs compilations pourront être réalisés par jour sans impactés les outils internes grâce à un isolement agile et fin entre infra de dev & infra de prod.

Suites aux mesures de performances, l'investissement mis en place pour réaliser ce projet sera amorti sur XT grâce à la scalabilité du SAAS pour déployer du nouveau contenu et de nouvelles fonctionnalités pour ses clients finaux. Plus le SAAS répondra aux besoins et plus l'intégrité du logiciel et de l'entreprise qui le conçoit sera importante. Par définition, le nombre d'utilisateur sera également croissant, voir exponentielle ce qui permettra d'exploiter tout le potentiel de la technologie devops mise en place et d'amortir les couts du projet.

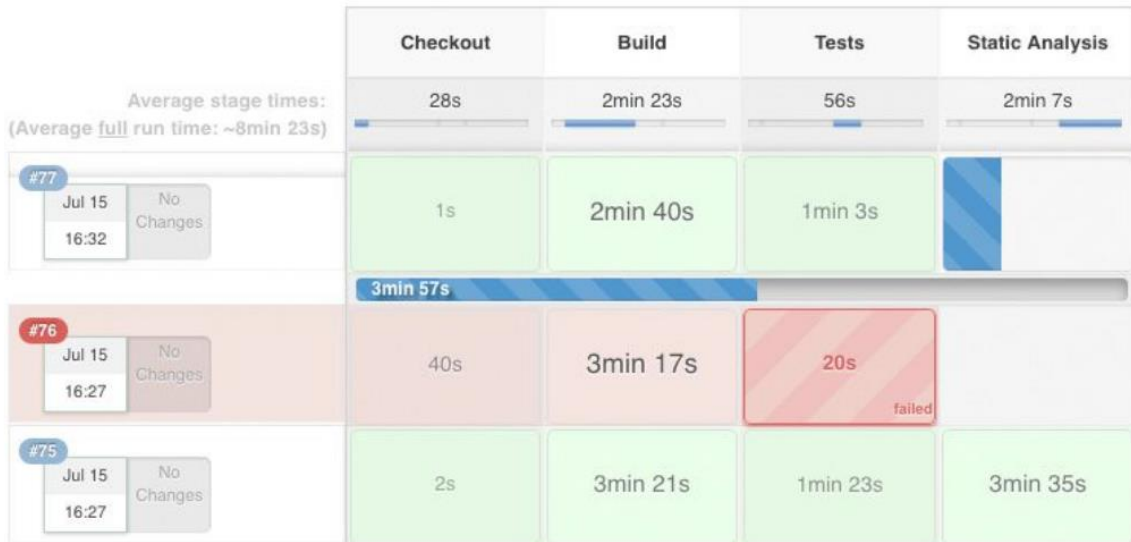
Les risques encourus par rapport à la pérennité futur du SAAS donne raison à l'évolution du SAAS vers IAAS.

Un bilan comptable et un calcul des couts par le contrôle de gestion pourra également être envisagé.

Annexes technique :

```
1 stages:
2   - build
3   - test
4   - quality
5   - package
6
7 cache:
8   paths:
9     - .m2/repository
10  key: "${CI_JOB_NAME}"
11
12 build_job:
13   stage: build
14   script:
15     - ./mvnw compile
16       -Dhttps.protocols=TLSv1.2
17       -Dmaven.repo.local=${CI_PROJECT_DIR}/.m2/repository
18       -Dorg.slf4j.simpleLogger.log.org.apache.maven.cli.transfer.Slf4jMavenTransferListener=WARN
19       -Dorg.slf4j.simpleLogger.showDateTime=true
20       -Djava.awt.headless=true
21       --batch-mode --errors --fail-at-end --show-version -DinstallAtEnd=true -DdeployAtEnd=true
22   image: openjdk:8-alpine
23
24 test_job:
25   stage: test
26   script:
27     - ./mvnw test
28       -Dhttps.protocols=TLSv1.2
29       -Dmaven.repo.local=${CI_PROJECT_DIR}/.m2/repository
30       -Dorg.slf4j.simpleLogger.log.org.apache.maven.cli.transfer.Slf4jMavenTransferListener=WARN
31       -Dorg.slf4j.simpleLogger.showDateTime=true
32       -Djava.awt.headless=true
33       --batch-mode --errors --fail-at-end --show-version -DinstallAtEnd=true -DdeployAtEnd=true
34   image: openjdk:8-alpine
35
36 code_quality_job:
37   stage: quality
38   image: docker:stable
39   allow_failure: true
40   services:
41     - docker:stable-dind
42   script:
43     - mkdir codequality-results
44     - docker run
45       --env CODECLIMATE_CODE="$PWD"
46       --volume "$PWD":/code
47       --volume /var/run/docker.sock:/var/run/docker.sock
48       --volume /tmp/cc:/tmp/cc
49       codeclimate/codeclimate analyze -f html > ./codequality-results/index.html
```

Stage View



All 16 Pending 0 Running 1 Finished 14 Branches Tags Run Pipeline Clear Runner Caches CI Lint

Status	Pipeline	Commit	Stages	Time
@ running	#52421235 by latest	Y master -> f2eb71d6 Add .gitlab-ci.yml	build: running	
@ passed	#52412574 by	Y master -> c3bba3d6 Update .gitlab-ci.yml	✓ ✓ ✓	00:14:34 35 minutes ago
@ passed	#52406248 by	Y master -> 10642a05 Update .gitlab-ci.yml	✓ ✓ ✓	00:15:00 1 hour ago
@ failed	#52405569 by yamllint invalid error	Y master -> 91b2b3dd Update .gitlab-ci.yml		1 hour ago
@ failed	#52405291 by yamllint invalid error	Y master -> a1c2c6bc Update .gitlab-ci.yml		1 hour ago
@ passed	#52145049 by	Y master -> 3d5c16e8 Update .gitlab-ci.yml	✓ ✓	00:11:03 2 days ago
@ passed	#52068512 by	Y master -> c62c4c50 Update .gitlab-ci.yml	✓ ✓	00:05:40 3 days ago
@ failed	#52067056 by	Y master -> e156a27d Update .gitlab-ci.yml	✓ ✗	00:02:39 3 days ago
@ failed	#52066499 by yamllint invalid error	Y master -> 05988a6d Update .gitlab-ci.yml		3 days ago
@ failed	#52066060 by	Y master -> ac8de787 Update .gitlab-ci.yml	✓ ✗ »	00:03:24 3 days ago
@ failed	#52063322 by	Y master -> 48ac16b2 Update .gitlab-ci.yml	✓ ✗ »	00:05:37 3 days ago

La page Pipeline avec toutes les exécutions du pipeline