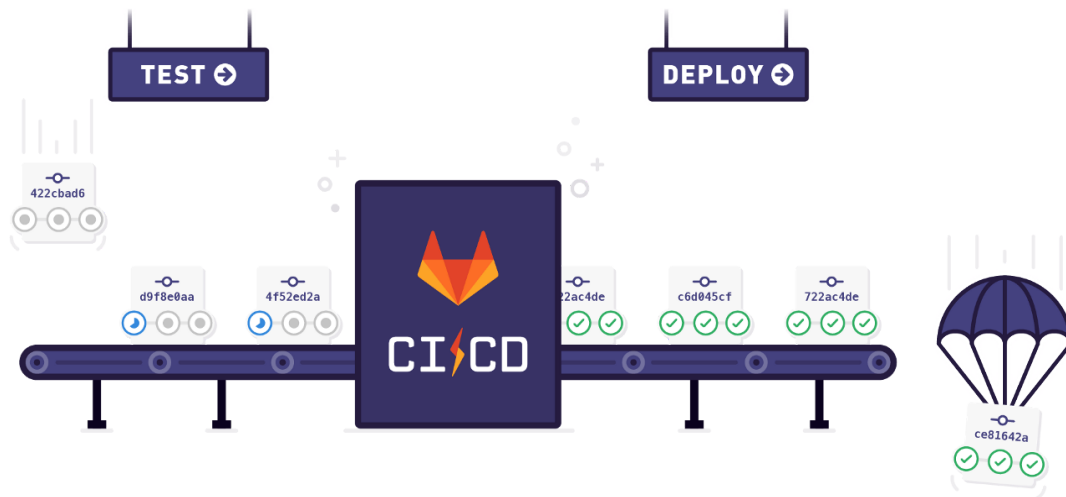


TP5.1_ Gitlab CI/CD



Objectifs du TP

- Installer Docker et préparer l'environnement de déploiement via la pipeline CI/CD.
- Déployer une image docker basique du docker hub via la pipeline CI/CD.
- Utilisation des variables sur la pipeline CI/CD.
- Configuration du déclenchement manuel d'une Pipeline CI/CD.

Prérequis

Avant de commencer ce TP, vous devez avoir satisfait les prérequis suivants

- Vous avez un projet sur GitLab sur lequel on va créer la pipeline CI/CD.
- Vous avez le rôle de maintainer sur votre projet.
- Vous avez validé le TP précédent (TP5.0 Create and run your first GitLab CI/CD pipeline).
- Vous avez installé un runner sur une VM linux.

Niveau de difficulté : Intermédiaire

Mode: En autonomie

Les étapes à suivre

A. Preparer le projet

- 1) Créez un fichier `.gitlab-ci.yml` à la racine de votre repository. C'est dans ce fichier que vous définissez les tâches CI/CD.

Rappel: Lorsque vous faites un commit sur ce fichier, le runner exécute vos tâches décrites dans le fichier `.gitlab-ci.yml`. Les résultats des travaux sont affichés dans un pipeline.

- 2) Assurez-vous que vous disposez de votre runner (état actif) pour exécuter vos tâches, désactivez les runners de type shared sur les paramètres de la CI/CD pour ce projet.

Si vous ne l'avez pas encore fait

1. Installer [GitLab Runner](#) sur la machine virtuelle.
2. [Register the runner](#) sur les paramètres du projet. Choisir l'exécuteur Docker ou shell.

B. Les jobs (tâches) à mettre sur la pipeline CI/CD

Veuillez compléter votre fichier **.gitlab-ci.yml** avec les tâches décrites comme suit:

- 1) Créez trois stages nommés respectivement: Prepare, Verify, Deploy.
- 2) Stage Prepare:
 - Afficher votre nom/prénom avec la commande `echo`.
 - Afficher le nom du job courant (Prepare) avec la commande `echo`.
 - Installer docker sur la machine via ce job.
- 3) Stage Verify:
 - Afficher le nom du job courant (Verify) avec la commande `echo`.
 - Exécuter des commandes docker pour afficher la version et lister les images disponibles s'il y'en a.
 - Effectuer un `docker system prune` pour nettoyer la machine.
- 4) Stage Deploy:
 - Lancer une image docker du docker hub (pour lancer l'image utilisez la cmd: `docker run hello-world`).
 - Spécifier explicitement que le déclenchement de ce job va être manuel.
 - Afficher la variable qui indique que le job a été déclenché manuellement en utilisant la commande `echo`.

C. Déclenchement de la pipeline et vérification des résultats

- 1) Commit du fichier **.gitlab-ci.yml**.
- 2) Consultez l'état de votre pipeline et de vos travaux (jobs).
- 3) Si l'un des job à un état **Failed** il faut vérifier les logs et debugger en ouvrant le terminal de chaque job et observer les résultats et output respectifs, ensuite relancer les jobs.
- 4) Lancer manuellement le job Deploy et s'assurer du bon déploiement de l'image.
- 5) Prenez des captures d'écran de l'état sain de votre pipeline et des résultats et outputs de chaque job.

Good Luck!