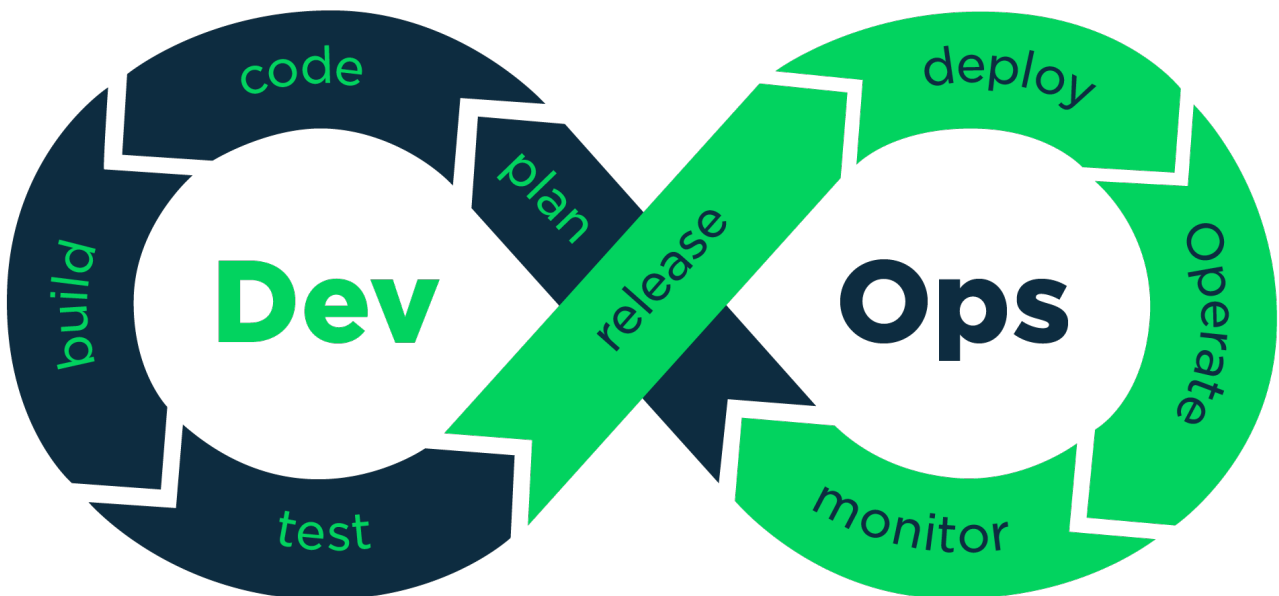


WorkShop Innovation n°1 autour de DEVOPS - le 8/12/2021

WorkShop Innovation n°1 autour de DEVOPS - le 8/12/2021

Prérequis si vous souhaitez manipuler (voir les liens ci-dessous "Copyright") :.....	1
Ordre du jour	2
DevTeams AGILE (tableau KANBAN sur gitlab)	2
Conteneurisation des environnements locaux de développement (Onboarding/ticket d'entrée réduit)	2
CI/CD	2
Serious Games pour la pratique	2
Docker VS Kubernetes	3
Services mesh	3
Questions (FeedBack).....	5
© Copyright :	6



Prérequis si vous souhaitez manipuler (voir les liens ci-dessous "Copyright") :

NOTE

1. GIT

2. DOCKER
3. DOCKER-COMPOSE (facultatif)
4. VISUAL STUDIO

Ordre du jour

DevTeams AGILE (tableau KANBAN sur gitlab)

Tableau d'ISSUES Gitlab ou JIRA

Conteneurisation des environnements locaux de développement (Onboarding/ticket d'entrée réduit)

Docker ou docker-compose par l'exemple

CI/CD

Intégration vs. livraison vs. déploiement continu

L'intégration, la livraison et le déploiement continu sont des pratiques qui visent à accélérer le processus de publication des logiciels en raccourcissant les boucles de feedback et en automatisant les tâches répétitives. Ces pratiques jouent un rôle clé dans la réalisation du principe Agile visant à livrer fréquemment aux utilisateurs des logiciels fonctionnels et utiles.

NOTE

- Les runners
- Les variables
- Les templates CI/CD
- Les artefacts
- Les ancrs
- L'héritage
- Conditions : only / except & rules

Serious Games pour la pratique

Les 3 applications de démonstration ci-dessous sont constituées de 3 socles techniques différents :

NOTE

1. <https://gitlab.univ-lille.fr/perso/pocs/m2-ili-war-maven-project.git>
2. <https://gitlab.univ-lille.fr/perso/pocs/m2-ili-war-gradle-project.git>
3. <https://gitlab.univ-lille.fr/perso/pocs/symfony5-docker-docker-compose-project.git>

IMPORTANT

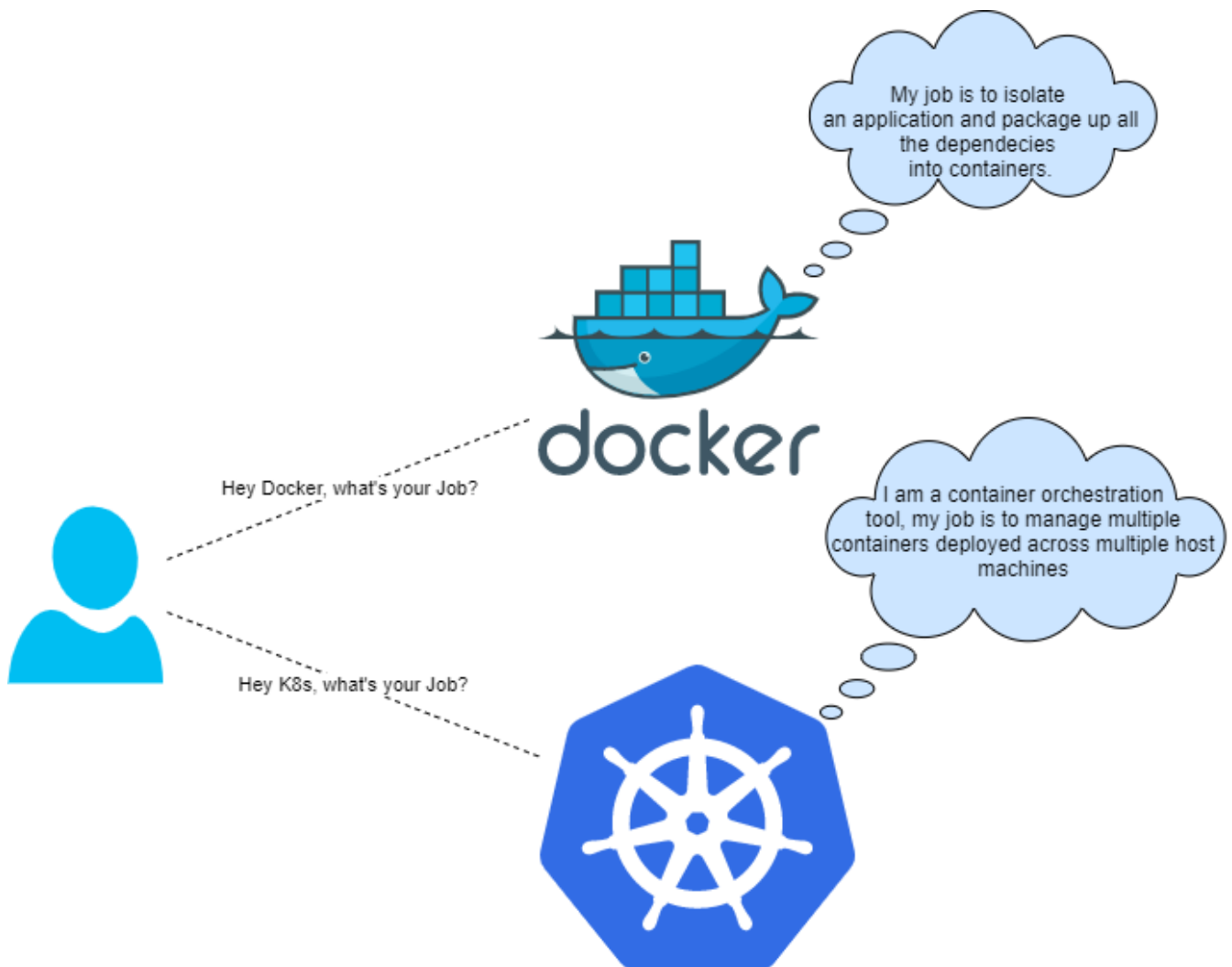
La quasi totalité des jobs sont définis dans des templates réutilisables (issus d'un autre projet)

<https://gitlab.univ-lille.fr/urbanisation/devops-project.git>

IMPORTANT

Une fois votre webapps orchestrée (déploiement continu ok) en environnement de staging (qualification), je vous montre l'accès à ses logs...

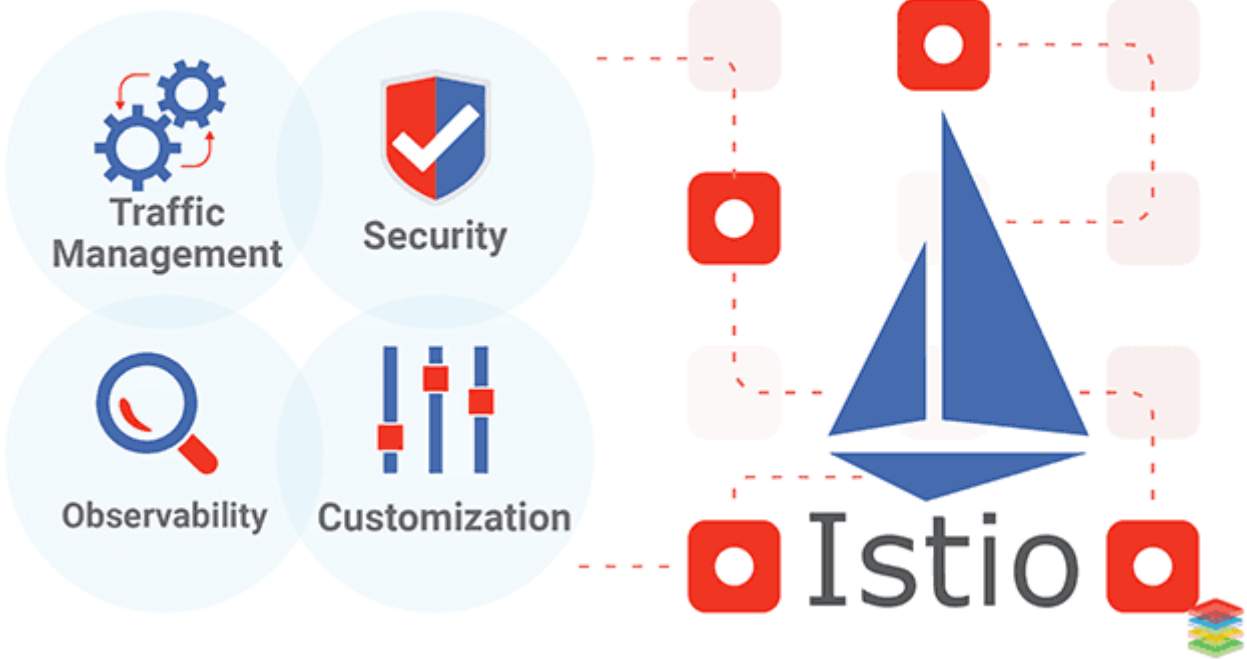
Docker VS Kubernetes

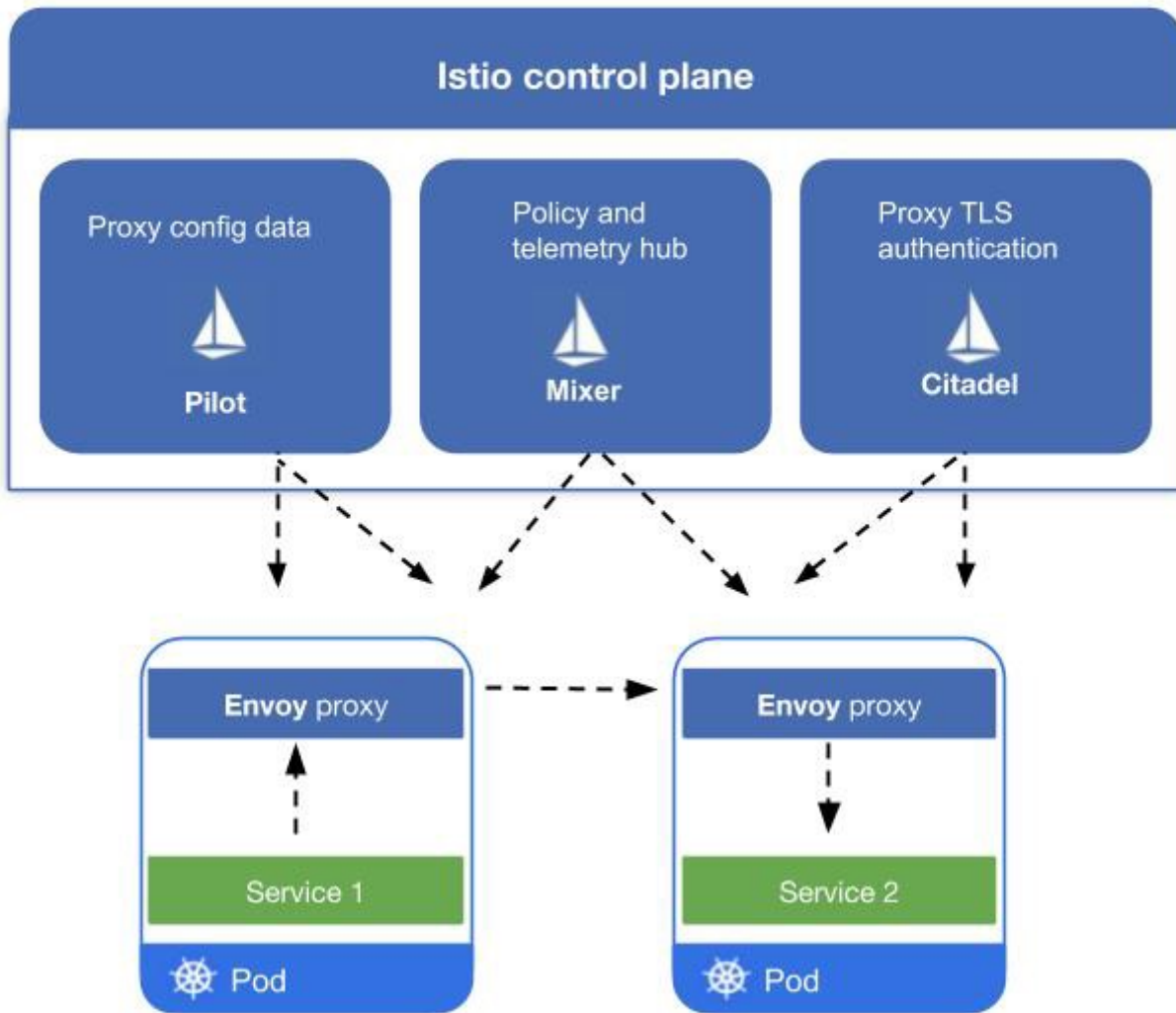


Docker est utilisé pour isoler votre application dans des conteneurs, tandis que Kubernetes est un outil de planification / orchestration de conteneurs et utilisé pour déployer et mettre à l'échelle votre application en gérant plusieurs conteneurs déployés sur plusieurs machines hôtes.

Services mesh

Les services mesh assurent l'interconnexion et le pilotage sécurisé des architectures principalement de type micro-services.





Questions (FeedBack)...



© Copyright :

- <https://www.jetbrains.com/fr-fr/teamcity/ci-cd-guide/>
- <https://docs.gitlab.com/ee/ci/yaml/>
- https://helm.sh/fr/docs/intro/using_helm/
- <https://geekflare.com/fr/docker-vs-kubernetes/>
- <https://www.docker.com/>
- <https://docs.docker.com/compose/>
- <https://git-scm.com/>
- <https://code.visualstudio.com/>