

Le déploiement

Table des matières

1. Le déploiement
2. Environnement/infrastructure de production
3. Les différents environnements
4. Les stratégies de déploiement
5. Procédure d'installation avec un installateur
6. Application à 12 facteurs
7. Facteurs clés de succès
8. Les questions à se poser
9. Les étapes du déploiement

1. Le déploiement

- On parle de déploiement informatique pour la mise en œuvre d'un nouveau système, un logiciel.
- C'est également le déploiement d'un réseau d'entreprise, d'un serveur de messagerie, d'outils collaboratifs, ou d'une application métier.
- Déployer le dernier système/logiciel/progiciel pour avoir plus de fonctionnalité s'inscrit dans le cadre d'un déploiement informatique.

1. Le déploiement

Objectif

Mettre en production l'application validée lors de la recette.

Les enjeux sont de :

- Mettre à disposition une solution stable et conforme à l'attendu.
- Accompagner les utilisateurs dans leur montée en compétence et dans l'appropriation de l'outil

Environnement infrastructure de production

2. Environnement/infrastructure de production

- Les Operations (ops) : personnes responsables de la mise en service d'une application
 - ✓ SysAdmin,
 - ✓ DBA,
 - ✓ Administrateurs réseaux, etc...)

Opération : Veut dire exploitation

2. Environnement/infrastructure de production

Firewall	Proxy	Load balancer	Web Server	Serveur d'application (conteneur, interpréteur)	BDCluster =>réplications de BDD	Serveur de BDD
F5	F5 HAProxy	F5 HAProxy	Apache, nginx, IIS	Tomcat, glassfish pour java mod_php et PHP-FPM pour php uWSGI, Gunicorn pour python, node, deno pour js		mysql, MariaDB SQL server Postgress, Oracle, etc...

2. Environnement/infrastructure de production

- **Firewall (pare-feu)** : logiciel et/ou un matériel permettant de faire respecter la politique de sécurité du réseau,
Il surveille et contrôle les applications et les flux de données
- **Serveur Proxy** : intermédiaire qui sépare les utilisateurs des sites Web sur lesquels ils naviguent. Les serveurs proxy assurent différents niveaux de fonctionnalité, de sécurité et de confidentialité, selon le type d'utilisation, les besoins ou la politique de l'entreprise.
- **Load Balancer** : répartiteur de charges entre plusieurs instances de container ou webserver

2. Environnement/infrastructure de production

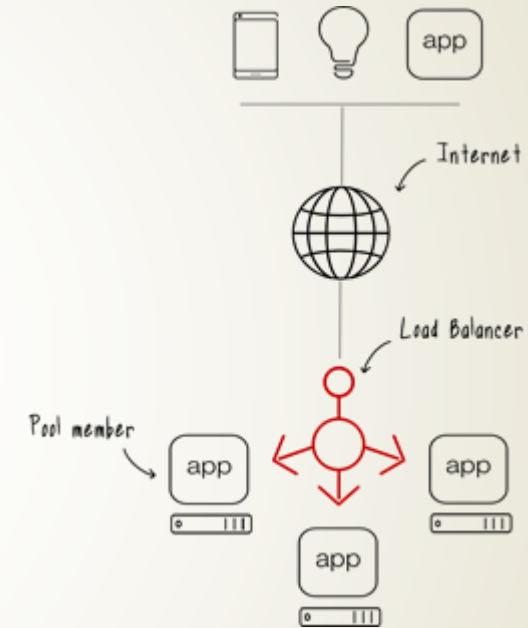
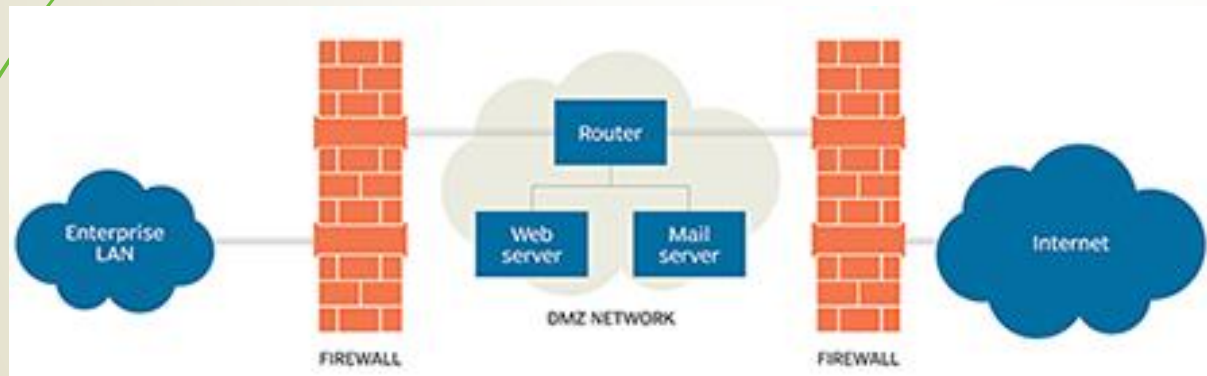
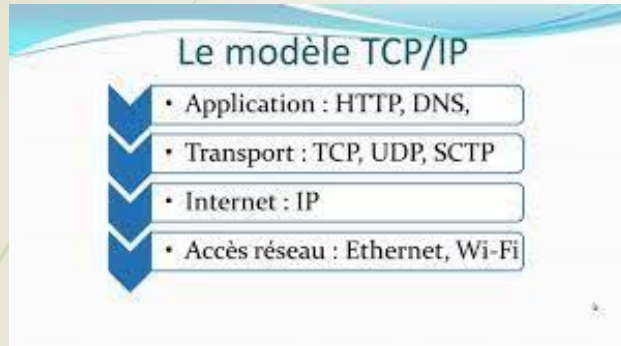
- **Webserver** : en frontal du conteneur.
 - ✓ Plus souples que le conteneur et plus rapide (tous les assets peuvent être sur ce serveur)
 - ✓ La conversion HTTPS/HTTP est faite par le Webserver ou avant le Webserver.

2. Environnement/infrastructure de production

➤ Conteneur :

- ✓ Peut faire tourner plusieurs applications dans la même VM (virtual machine) avec le même espace mémoire
- ✓ Peut déléguer des rôles comme un objet httpRequest, les connexions BDD, etc...
- ✓ Ils discutent entre eux avec un protocole réseau de modèle TCP/IP (Transmission Control Protocol/Internet Protocol)

2. Environnement/infrastructure de production



Les différents environnements

3. Les différents environnements



3. Les différents environnements

Développement (DEV) :

- Dans l'environnement de Développement,
 - ✓ Les développeurs créent et déploient du code dans un laboratoire où l'application est testée au niveau le plus élémentaire.
 - ✓ Lorsque l'application satisfait certains critères de qualité, elle passe à l'environnement suivant :
 - Machine Locale
 - Eventuellement un environnement de développement commun

3. Les différents environnements

Test d'Intégration Système (SIT)

- L'application est testée pour garantir qu'elle fonctionne avec les applications et systèmes existants.
- Lorsque l'application réussit les tests d'intégration et les tests systèmes, elle est déployée dans l'environnement suivant.

3. Les différents environnements

Tests d'acceptation des utilisateurs (UAT : User Acceptance Testing) :

- L'application est testée manuellement par des personnes de la MOA (maîtrise d'ouvrage) pour s'assurer qu'elle fournit les fonctionnalités requises pour les utilisateurs finaux.
- Cet environnement est généralement proche de celui de la production, notamment pour les données.
- Lorsque l'application répond à ces exigences, elle est promue vers l'environnement final.

3. Les différents environnements

Production (PROD) :

- L'application est mise à la disposition des utilisateurs.
- L'analyse du comportement utilisateur de l'application est obtenue en surveillant la disponibilité et la fonctionnalité de l'application.
- Tous les mises à jour ou correctifs sont introduits dans l'environnement DEV et suivent le même cycle.

Les stratégies de déploiement

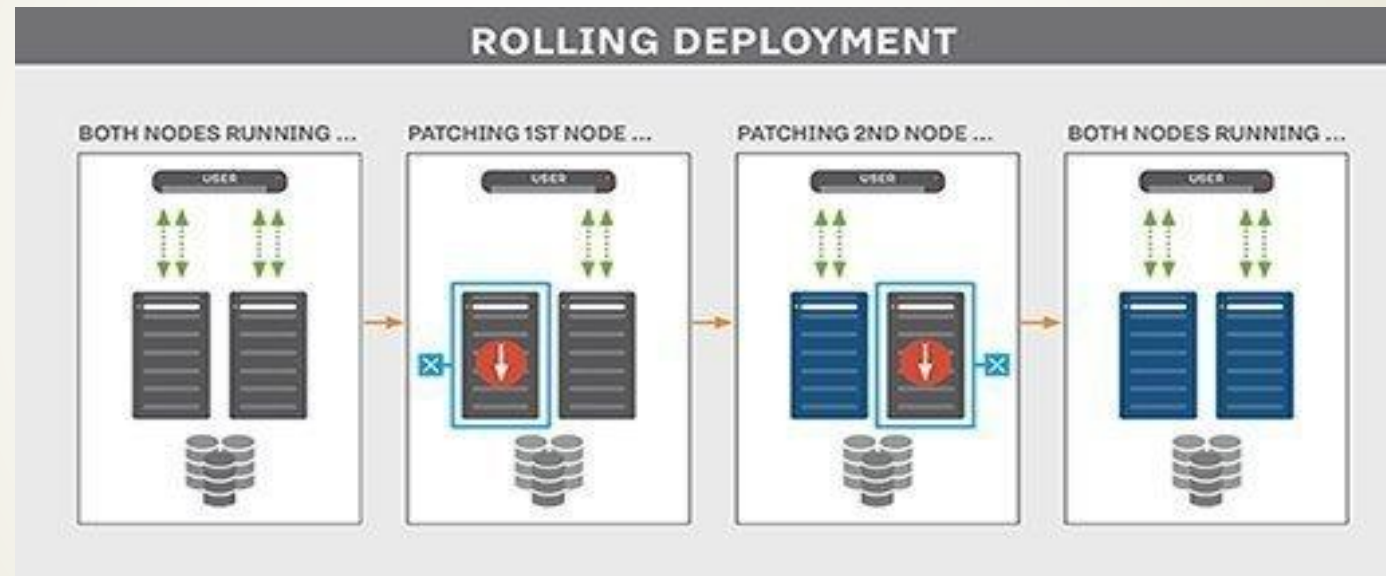
4. Les stratégies de déploiement

Déploiement progressif

- Pour les environnements de production ayant une application distribuée. Ceux-ci se composent de plusieurs serveurs ou instances en cloud, chacun hébergeant une copie de l'application
- Echelonnement des modifications
 - ✓ La mise à jour s'active sur certains serveurs ou instances avant d'autres.
 - ✓ Les premiers cités exécutent la nouvelle version de l'application, tandis que d'autres continuent à héberger l'ancienne

4. Les stratégies de déploiement

Déploiement progressif



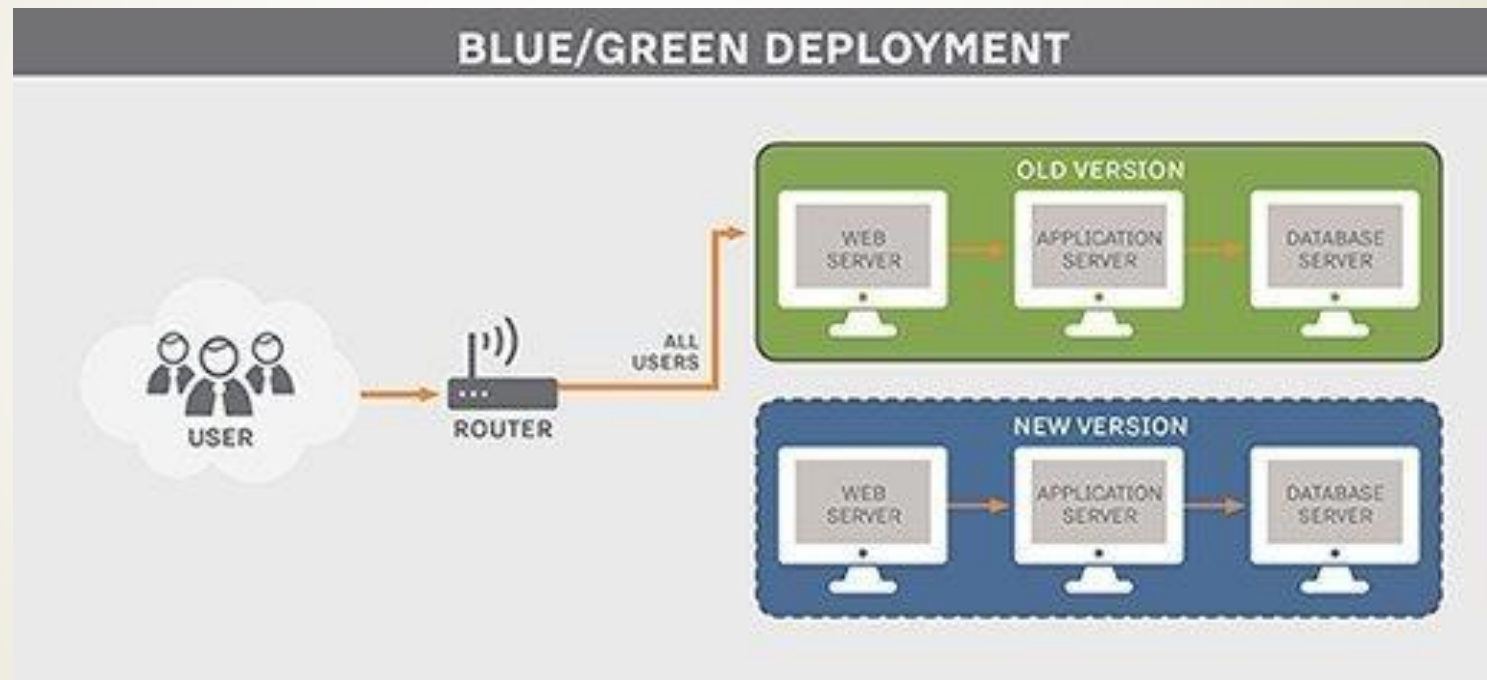
4. Les stratégies de déploiement

Déploiement blue/green

- Deux infrastructures d'hébergement d'applications distinctes : une bleue et une verte
- À tout moment, l'une de ces configurations d'infrastructure héberge la version de production de l'application (la verte), tandis que l'autre est tenue en réserve (la bleue)
- Une fois qu'ils ont suffisamment testé le déploiement, les administrateurs peuvent échanger le trafic d'une infrastructure à l'autre

4. Les stratégies de déploiement

Déploiement blue/green



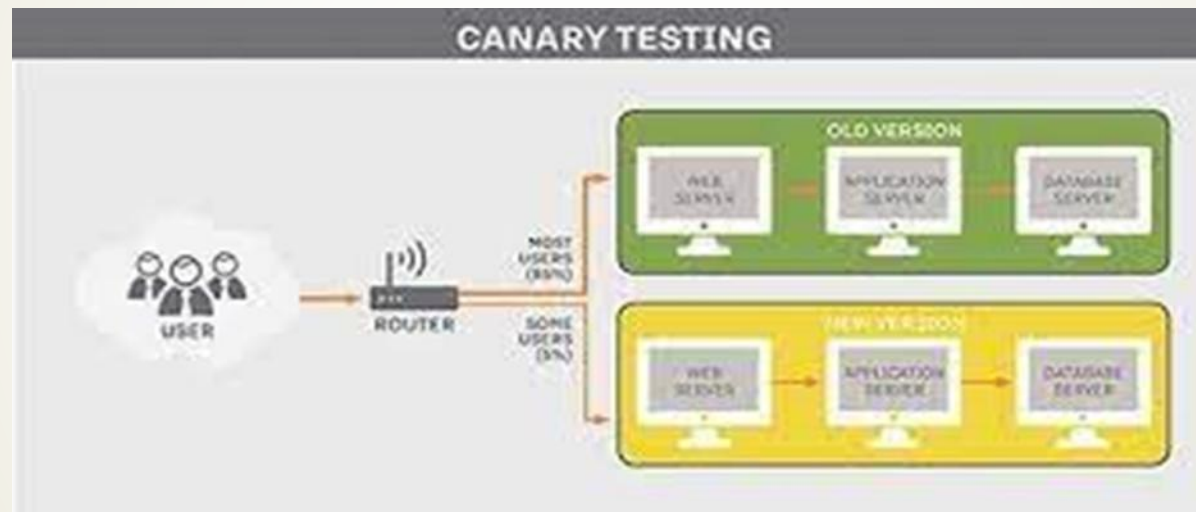
4. Les stratégies de déploiement

Déploiement canary (ou canari)

- Similaire à un déploiement progressif
- Mais seuls certains utilisateurs reçoivent l'accès à la nouvelle version de l'application, plutôt que certains serveurs
- Le sous-ensemble d'utilisateurs spécifié, c'est-à-dire les canaris, trouve les faiblesses et améliore la mise à jour avant que l'équipe informatique ne la déploie auprès de tous les utilisateurs.
- La répartition est souvent de 5% de canaris

4. Les stratégies de déploiement

Déploiement canary (ou canari)



<https://www.lemagit.fr/conseil/DevOps-les-trois-types-de-deploiement-les-plus-populaires>

Procédure d'installation avec un installateur

5. Procédure d'installation avec un installateur

Un installateur (ou installeur) désigne un programme d'installation permettant d'installer un logiciel

- Ces programmes interagissent généralement avec l'utilisateur à l'aide de boîtes de dialogue.
- Leur but est de procéder automatiquement à la copie sur l'ordinateur cible (et parfois au paramétrage) des programmes et fichiers composant le logiciel, afin d'assurer son bon fonctionnement.

5. Procédure d'installation avec un installateur

Pour réaliser une procédure d'installation d'une application, il faudra décomposer celle-ci en trois parties

- Fabriquer une version finale (release) de l'application (make) en édition indépendante (statique) ou non (dynamique)
- Fabriquer un installateur pour l'application (setup)
- Déployer cette application sur une machine

Application à 12 facteurs
6 sont à appliquer : les 5 premiers + le
dixième

<https://12factor.net>

<https://12factor.net/fr/>

6. Application à 12 facteurs

I. Base de code

- Une base de code suivie avec un système de contrôle de version
 - ✓ Une *base de code* correspond à chaque dépôt (dans un système de contrôle de version centralisé tel que Subversion), ou tout ensemble de dépôts qui partage un commit racine (dans un système de contrôle de version décentralisé comme Git)
 - ✓ un seul dépôt de code source versionné = une seule application

6. Application à 12 facteurs

II. Dépendances

- Déclarez explicitement et isolez les dépendances (Maven, npm, composer, etc...) en les récupérant en entier et sans les modifier
- ✓ Installation de l'application simplifiée pour les développeurs

6. Application à 12 facteurs

III. Configuration

- Stockez la configuration dans l'environnement => on doit exporter le même bundle de l'environnement DEV vers l'UAT puis vers la prod. On ne doit travailler qu'avec les variables d'environnement. (Rq : Maven ne suit pas cette recommandation)

La configuration d'une application est tout ce qui est susceptible de varier entre les environnements (environnement de développement, validation, production, etc.)

- ✓ Les ressources gérées par la base de données,
- ✓ Les identifiants pour des services externes, tel qu'Amazon S3 ou Twitter
- ✓ Les valeurs spécifiques au déploiement, tel que son nom d'hôte canonique

6. Application à 12 facteurs

IV. Services externes

Traitez les services externes comme des ressources attachées

- ✓ Un *service externe* (backing service) correspond à tout service que l'application utilise à travers le réseau pour son fonctionnement nominal.
- ✓ On doit pouvoir changer les services sans modifier le code
- ✓ Cela concerne par exemple les bases de données les systèmes de messages/files, les services SMTP pour l'envoi d'email ainsi que les systèmes de cache et même les services que l'on peut consommer à travers une API (comme Twitter, Google Maps..)

6. Application à 12 facteurs

V. Assemblez, publiez, exécutez

Séparez strictement les étapes d'assemblage et d'exécution

Une base de code est transformée en un déploiement (non-développement) à travers les étapes suivantes :

- **L'étape d'assemblage (ou "build")**
- ✓ Transformation qui convertit un dépôt de code en un paquet autonome exécutable appelé l'assemblage (ou "build").
- ✓ En utilisant une version du code référencée par un commit spécifié lors du processus de déploiement, l'étape d'assemblage va chercher toutes les dépendances externes et compile les fichiers binaires et les ressources.

6. Application à 12 facteurs

V. Assemblez, publiez, exécutez

- **L'étape de publication (ou "release")**
 - ✓ Prend l'assemblage produit à l'étape précédente et le combine avec la configuration de déploiement courante.
 - ✓ La release résultante contient à la fois l'assemblage et la configuration, et elle est prête pour une exécution immédiate dans l'environnement d'exécution.

6. Application à 12 facteurs

V. Assemblez, publiez, exécutez

➤ L'étape d'exécution (ou "runtime")

Fait fonctionner l'application dans l'environnement d'exécution, en lançant l'ensemble des processus de l'application associée à la release considérée.

6. Application à 12 facteurs

X. Parité dev/prod

Gardez le développement, la validation et la production aussi proches que possible

- Historiquement, il y a eu un fossé conséquent entre le développement (un développeur qui fait des modifications sur un déploiement local de l'application) et la production (un déploiement de l'application accessible aux utilisateurs finaux)

6. Application à 12 facteurs

X. Parité dev/prod

Ce fossé se manifeste de trois manières :

- ✓ **Le fossé temporel** : un développeur peut travailler sur du code qui peut prendre des jours, des semaines ou des mois avant d'aller en production
- ✓ **Le fossé des personnes** : les développeurs écrivent le code, et d'autres personnes le déploient.
- ✓ **Le fossé des outils** : les développeurs peuvent utiliser SQLite, et un OS X, alors que le déploiement de production utilise Oracle, et Linux par exemple.

Facteurs clés de succès

7. Facteurs clés de succès

Formaliser et communiquer le périmètre, l'organisation et le planning du déploiement

Les enjeux sont de :

- Clarifier les responsabilités des différents acteurs : le qui fait quoi, quand.
- Identifier les risques et les leviers pour maîtriser les délais et les coûts.

7. Facteurs clés de succès

Construire un planning réaliste

- Un planning bien construit tient compte des délais incompressibles entre les différentes phases, des responsabilités des acteurs et comprend des marges suffisantes pour absorber les aléas.
- Partir de la date souhaitée de démarrage pour construire un retroplanning peut être une option mais uniquement si elle conduit à des objectifs tenables.

7. Facteurs clés de succès

Anticiper et gérer les risques

- Chaque risque identifié doit être noté et une solution envisagée pour y palier.
- Votre rôle sera de suivre ces risques à chaque étape du déploiement et d'adapter les solutions envisagées lorsque ces risques surviendront.

7. Facteurs clés de succès

Définir le rôle de chacun

- Un Déploiement est un travail d'équipe, vous devez vous entourer des acteurs indispensables à sa réalisation et les coordonner.
- Pour cela, identifiez bien les différentes tâches et la durée nécessaire pour les accomplir afin de définir le profil des personnes qui devront les réaliser.

7. Facteurs clés de succès

Investir les temps de Formation

ENTRE 15 JOURS AVANT ET 15 JOURS APRÈS LE DÉPLOIEMENT

- Trop tôt, vos utilisateurs ne se souviendront plus du contenu dispensé
- Trop tard, ils auront déjà été confrontés au logiciel.

Ce sont parfois les utilisateurs clés, formés avant la recette, qui formeront, à leur tour, les utilisateurs finaux.

7. Facteurs clés de succès

Réaliser un run à blanc

- Le run à blanc consiste en une répétition des différentes étapes de mise en production de l'application.
- Bien documenté, ce modus operandi permet de simuler le temps nécessaire et de planifier le déploiement pour respecter les délais et assurer la qualité de ce qui est mis en production.
- Les reprises de données et les interfaces peuvent impacter le temps de déploiement, le run à blanc permet de bien anticiper l'impact qu'elles auront lors du déploiement.

7. Facteurs clés de succès

Avoir le Go de la direction

- Pour garantir le soutien de la direction lors de la phase de déploiement, il est primordial d'organiser une réunion du comité de pilotage juste après la fin de la recette, qui permettra de décider si l'organisation est prête à déployer le logiciel tel qu'il a été recetté.

Les questions à se poser

8. Les questions à se poser

- Est-ce que la nouvelle organisation a bien été expliquée en amont du déploiement de l'outil ?
- Est-ce que les utilisateurs clés disposent d'une vision claire et unifiée des nouveaux processus ?
- Est-ce que la formation utilisateurs finaux a bien été planifiée ?

8. Les questions à se poser

- Quel est l'ordre des actions à réaliser pour déployer le logiciel ?
- Est-ce qu'un run à blanc est nécessaire et a été planifié pour simuler ces étapes ?
- Le procès-verbal de recette est-il signé ? La direction a-t-elle donné son go au déploiement ?

9. Les étapes du déploiement

