



# DÉVELOPPER ET TESTER VOS RÔLES ANSIBLE

Nicolas hicher

*[2017-11-22 Wed]*

# AGENDA

Intégration et livraison continues

Présentation de Software-Factory

Démo

Conclusion

Contacts

# INTÉGRATION ET LIVRAISON CONTINUES

# CONCEPTS

- *L'intégration continue est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée.*

# CONCEPTS

- *La livraison continue est une approche d'ingénierie logicielle dans laquelle les équipes produisent des logiciels dans des cycles courts, ce qui permet de le mettre à disposition à n'importe quel moment. Le but est de construire, tester et diffuser un logiciel plus rapidement.*

# BONNES PRATIQUES

- Utiliser un gestionnaire de version
- Transformer les procédures en code
- Revue de code en équipe
- Construire des tests automatiques de validation
- Mise en production rapide et automatisée

# UTILISER UN GESTIONNAIRE DE VERSION

- Dépôt centralisé
- Facilité de rollbacks
- Gestion de branches
- Historique des modifications



# TRANSFORMER LES PROCÉDURES EN CODE

- La connaissance orale devient explicite/écrite
- Précision et évolutivité par rapport aux opérations humaines
- Réduction ou suppression des ambiguïtés des procédures
- Simplification des procédures pour les équipes
- Gestion des infrastructures comme code



# REVUE DE CODE EN ÉQUIPE

- Réduction significative des erreurs
- L'équipe est mutuellement responsable des changements
- Faciliter les discussions pour les choix techniques
- Permet de garder un trace écrite du 'pourquoi' des changements, avec du contexte
- Permet aux membres juniors de l'équipe d'apprendre

# TESTS AUTOMATISÉS

- L'ajout de nouveaux tests est simplifié
- Tests immédiat des modifications
- Remontée rapide des erreurs
- Facilité pour les tests de régression, pour un cycle de développement rapide
- Améliore la qualité du code de manière incrémentale

# MISE EN PRODUCTION RAPIDE

- Réduire le coût, le temps et les risques associés a la livraison
- Le processus de déploiement doit être simple, répétable et idéalement automatisé
- Évolution incrémentale

# PRÉSENTATION DE SOFTWARE-FACTORY

# SOFTWARE-FACTORY

- Forge de développement libre
- Facilite le travail collaboratif
- Qualité de code grâce a l'intégration continue
- Inspiré du CI d'Openstack
- Installé et configuré via `Ansible`



# QUELQUES LOGICIELS PRÉSENTS SUR SOFTWARE-FACTORY

- **gerrit**
- **zuul3**
- **nodepool3**
- etherpad
- pasteit
- gitweb
- mumble
- logserver #(ELK)
- repxplorer



# GERRIT

- Outil de revue de code pour Git
- Les revues de code se font via l'interface web ou cli (gertty)
- Un correctif n'est intégré au code qu'après avoir reçu une revue favorable de la part du CI et des pairs



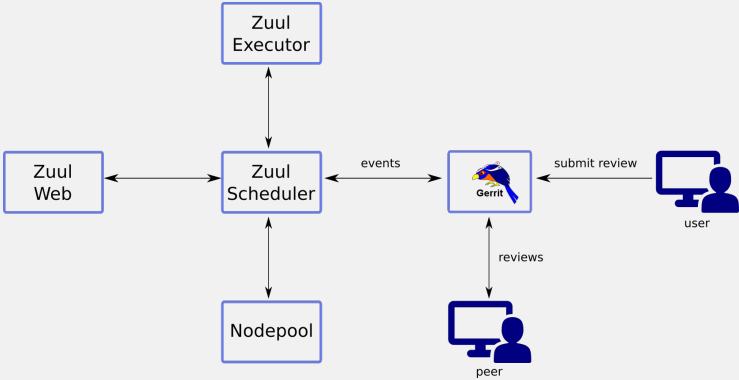
# ZUUL

- Système de "project gating"
- Développé par l'équipe d'infrastructure Openstack
- Tous les patchs proposés sont testés lors de la soumission et avant d'être mergé.
- Dépendances possible entre les projets pour les tests
- Les jobs sont exécutés en parallèle ou bien chaînés
- Les jobs sont écrits en Ansible





# WORKFLOW



DÉMO

# BESOINS

Pour cette demo, nous allons effectuer 3 tests avant d'intégrer un changement de code

- Execution du test linter pour valider la syntaxe
- validation du role sur `fedora 25`
- validation du role sur `fedora 26`

# ADMINISTRATEUR

L'administrateur doit gérer plusieurs aspects :

- Configurer le provider (ici `OpenStack`), l'utilisation de containers est également possible depuis la version courante de `Software-Factory` (2.7.0)
- Valider les changements proposés par les utilisateurs dans le config-repo (création de projets ou d'images)

# UTILISATEUR

L'utilisateur peut créer et configurer un projet:

- Définition du projet: dépôts Git, règles de contrôle d'accès (\*)
- Définition des jobs de test, publication, etc
- Définition des images consommées par les jobs (\*)
- Les jobs sont écrit en `Ansible`

Ces définitions se font sous forme de patches soumis au processus de revue de code:

- Les définitions marquées (\*) doivent être approuvées par l'administrateur
- la définition des jobs se fait via un fichier à la racine du dépôt du projet, il est géré par l'équipe de développement

CONCLUSION

# POUR RÉSUMER

- pour les utilisateurs, ajouter facilement des tests fonctionnels en quelques lignes d'ansible
- pour le responsable de l'équipe, réduire la pression sur les équipes, tout le monde est impliqué
- pour le responsable de livraison, augmenter la fréquence et la qualité des releases
- pour le responsable de l'infrastructure, augmenter la confiance dans les processus de mise à jour.

# CONTACTS



# CONTACTS:

`https://softwarefactory-project.io/sf/welcome.html`

- courriel: `softwarefactory-dev@redhat.com`
- irc: `#softwarefactory` (freenode)