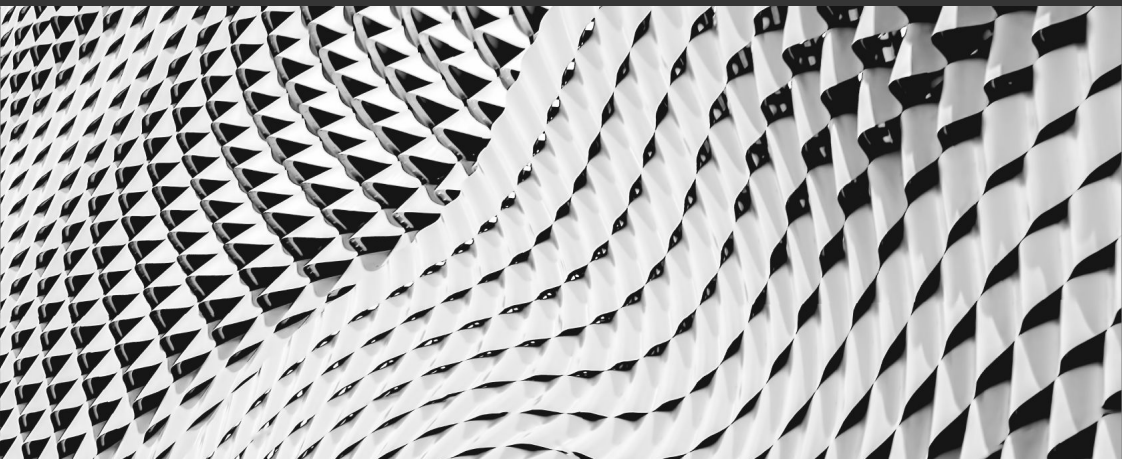


Introduction



- Intégration continue
- Livraison continue
- Ansible Tower

Intégration continue

Centralisation des informations (code, test,...)

Validation en continue

L'automatisation au centre des processus

Better

Un code stable

Faster

Rapidité de deployment

Stronger

Multitude de testes

Livraison continue

Livraison via dans des cycles courts

Déploiement automatisé

Simplification des déploiements

Better

Des déploiements stables

Faster

Des déploiements rapides

Stronger

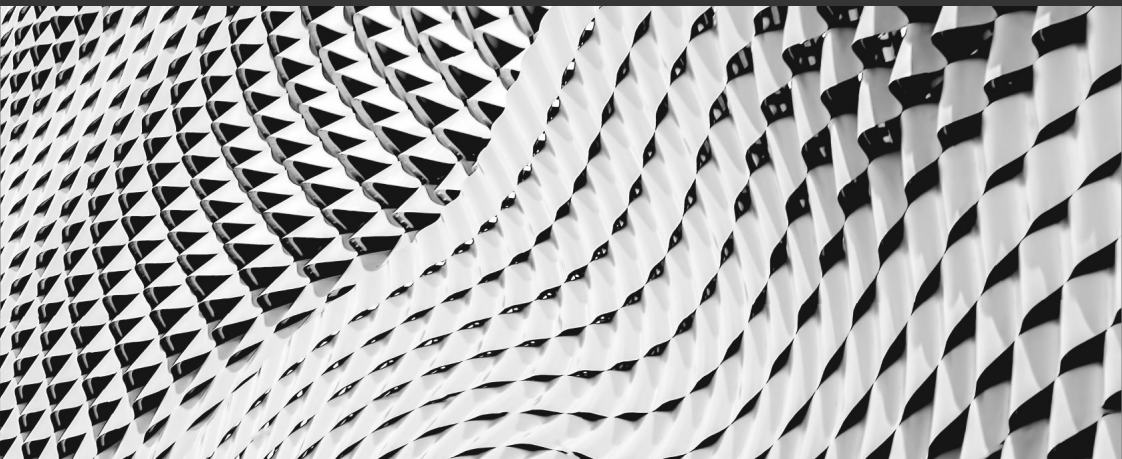
Réduit la peur de déployer

Ansible Tower

Centralise et automatise votre infrastructure informatique avec :

- un tableau de bord visuel
- une gestion graphique des ressources.
- un contrôle d'accès basé sur les rôles
- la planification des travaux
- des notifications intégrées
- une compatibilité REST API et CLI
- et bien plus encore.

Ansible Tower



- Template
- Workflow
- Webhook

Template

Un template est la définition d'un ensemble de paramètres permettant d'exécuter un job Ansible.

Il comprend :

- un playbook
- un inventaire
- des identifiants
- et autre.

Build

DÉTAILS PERMISSIONS NOTIFICATIONS JOBS COMPLÉTÉS PLANNINGS AJOUTER UN QUESTIONNAIRE

* NOM ME LE DEMANDER AU LANCEMENT

DESCRIPTION

* TYPE DE JOB ME LE DEMANDER AU LANCEMENT

* INVENTAIRE ME LE DEMANDER AU LANCEMENT

* PROJET ME LE DEMANDER AU LANCEMENT

* PLAYBOOK ME LE DEMANDER AU LANCEMENT

INFORMATIONS D'IDENTIFICATION ME LE DEMANDER AU LANCEMENT

FORKS

LIMITE ME LE DEMANDER AU LANCEMENT

* VERBOSITÉ ME LE DEMANDER AU LANCEMENT

BALISES DU JOB ME LE DEMANDER AU LANCEMENT

BALISES DE SAUT ME LE DEMANDER AU LANCEMENT

LIBELLÉS

GROUPES D'INSTANCES

JOB SLICING

TIMEOUT

AFFICHER LES MODIFICATIONS ME LE DEMANDER AU LANCEMENT

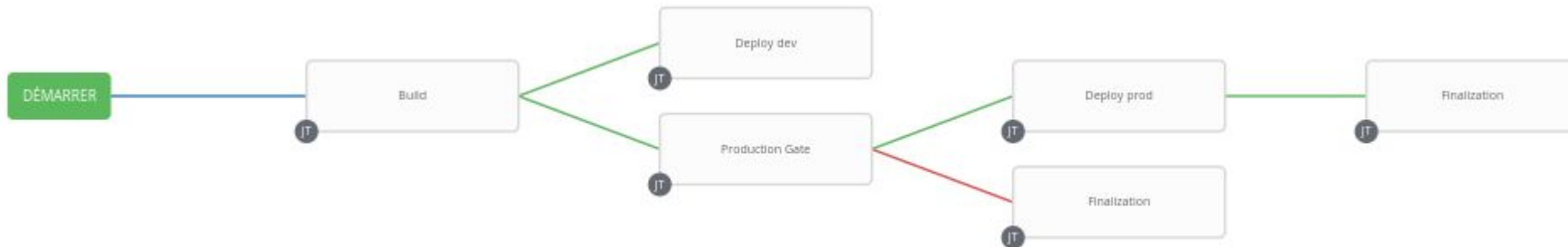
OPTIONS

- ACTIVER L'ÉLÉVATION DES PRIVILÈGES
- ENABLE PROVISIONING CALLBACKS
- ENABLE WEBHOOK
- ACTIVER LES JOBS PARALLÈLES
- ENABLE FACT CACHE

Workflow

Un workflow permet d'ordonner des actions. Ces actions peuvent aussi bien être des templates que des workflows.

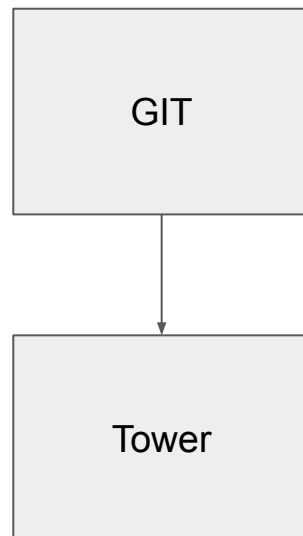
Dans Ansible Tower depuis la version 3.1



Webhook

Un Webhook offre la possibilité de recevoir et interagir à des événements venant d'un site de confiance.

Ansible Tower support aujourd'hui Github et Gitlab
Dans Ansible Tower depuis la version 3.6



```
1 tower_webhook_event_guid: 3af24ad0-9978-11ea-8b18-e4a9a8e04157
2 tower_webhook_event_ref: d446351ecd04370e1fdbd2884df2f5853612d560
3 tower_webhook_event_type: push
4 tower_webhook_payload:
5   after: d446351ecd04370e1fdbd2884df2f5853612d560
6   base_ref: refs/heads/master
7   before: '0000000000000000000000000000000000000000'
8   commits: []
9   compare: 'https://github.com/pierreblanc/web_content_source/compare/v1.4'
10  created: true
11  deleted: false
12  forced: false
13  head_commit:
14    added: []
15    author:
```

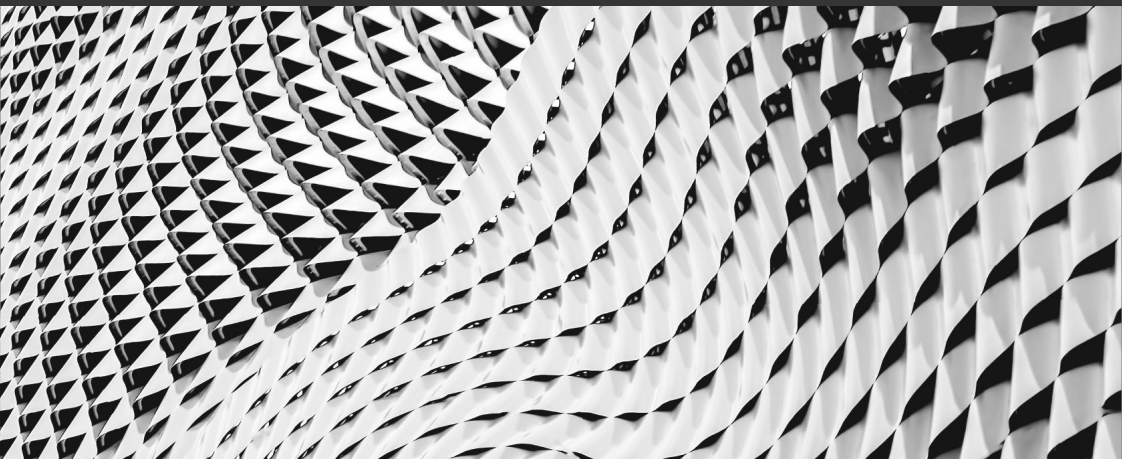
API vs Webhook

Les webhooks envoient des données prédéfinies, ils sont utilisés pour créer une communication entre Tower et un service de gestion de source

Les API offrent beaucoup plus de flexibilité, et peuvent être utilisés dans tous les autres cas.

```
TOWER API REST
API REST
API REST
GET /api/
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
X-API-Node: localhost
X-API-Time: 0.014s
{
  "description": "API REST AWX",
  "current_version": "/api/v2/",
  "available_versions": {
    "v2": "/api/v2/"
  },
  "oauth2": "/api/o/",
  "custom_logo": "",
  "custom_login_info": "",
  "login_redirect_override": null
}
```

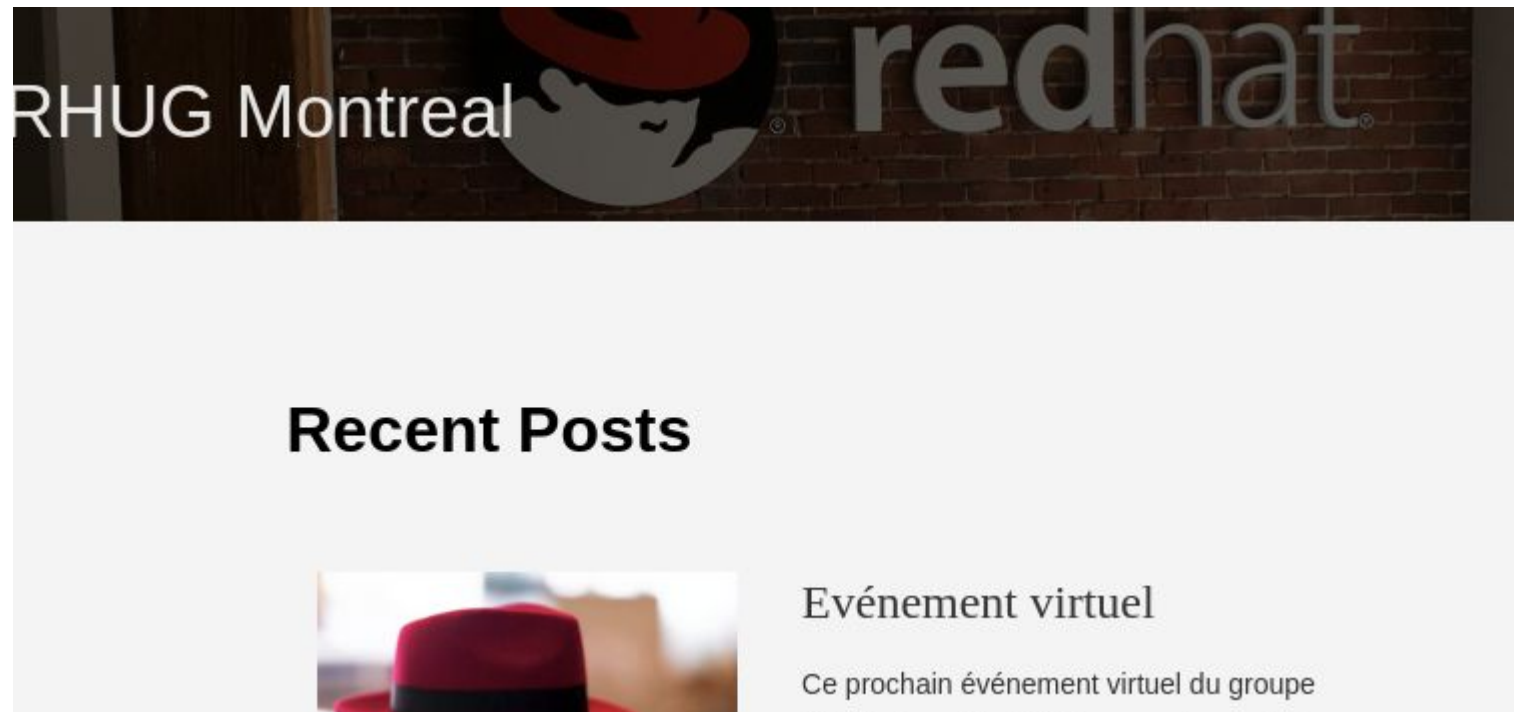
Mise en pratique



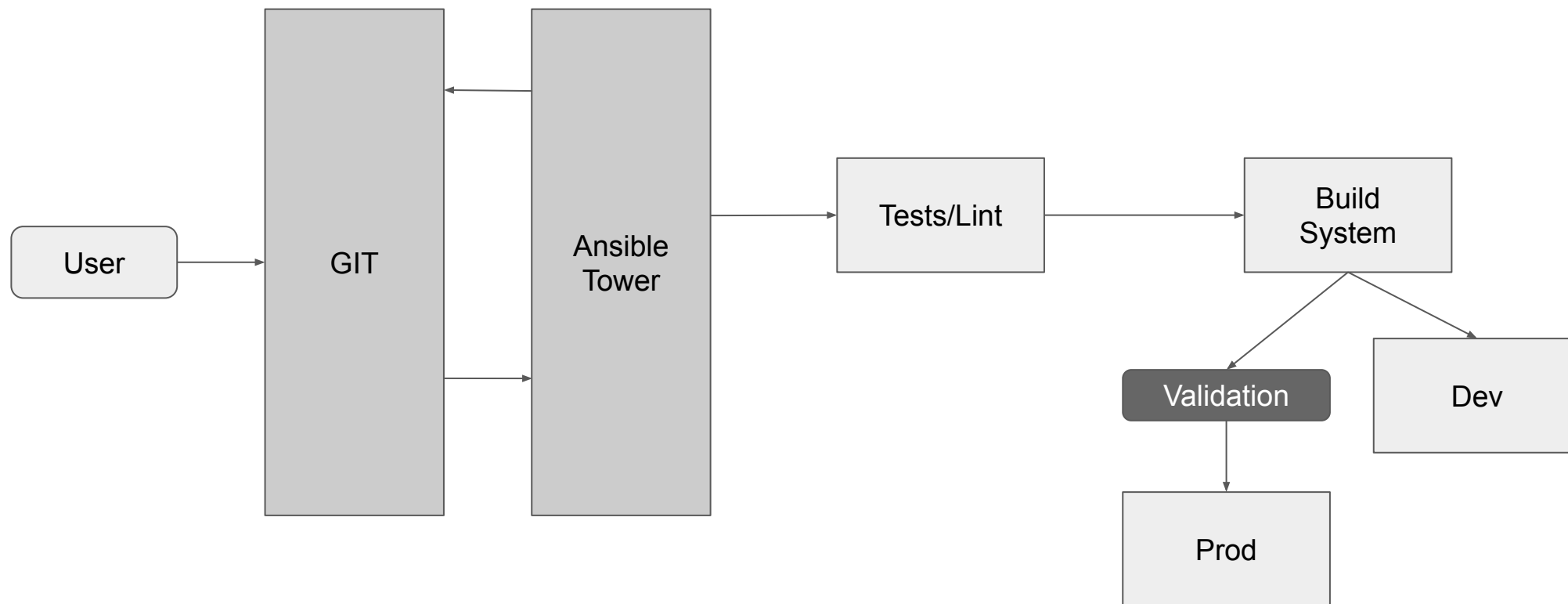
- Introduction
- Schéma
- Démonstration

Scénario

- Hébergement d'un site statique
- Plusieurs environnements
- Méthodologie Agile
- Equipe devops



Strategie



Merci !

Des questions ?

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 twitter.com/RedHat