



McGill

Information
Technology
Services

Automatiser 1000 sites web avec Ansible et Gitlab CI

Rencontre Ansible Montréal et Québec

7 juin 2023

**Connect.
Learn.
Innovate.**



Qui suis-je



Thomas Fline

Développeur Web

Université McGill (depuis 2013)

✉ thomas.fline@mcgill.ca

in <https://linkedin.com/in/thomasfline>

🐙 <https://github.com/fengtan>

💧 <https://drupal.org/u/fengtan>

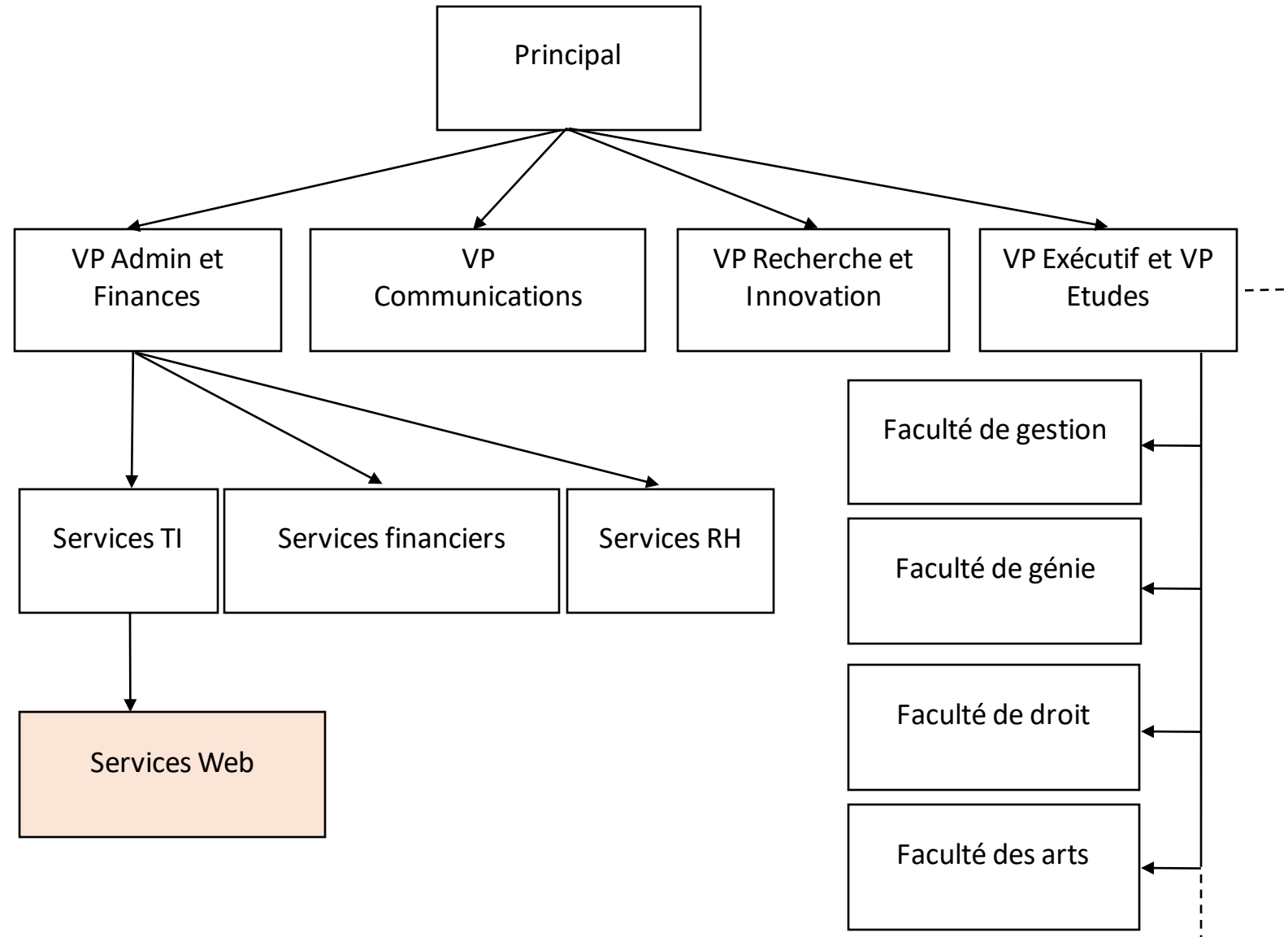
🌐 <https://www.mcgill.ca/it>

Equipe des Services Web de McGill

Equipe intégrée aux services TI centraux. Offre des sites web uniformes aux différents départements de l'Université.

10 personnes incluant:

- Gestionnaire
- Développeurs backend
- Développeurs frontend
- Analystes en soutien et communication

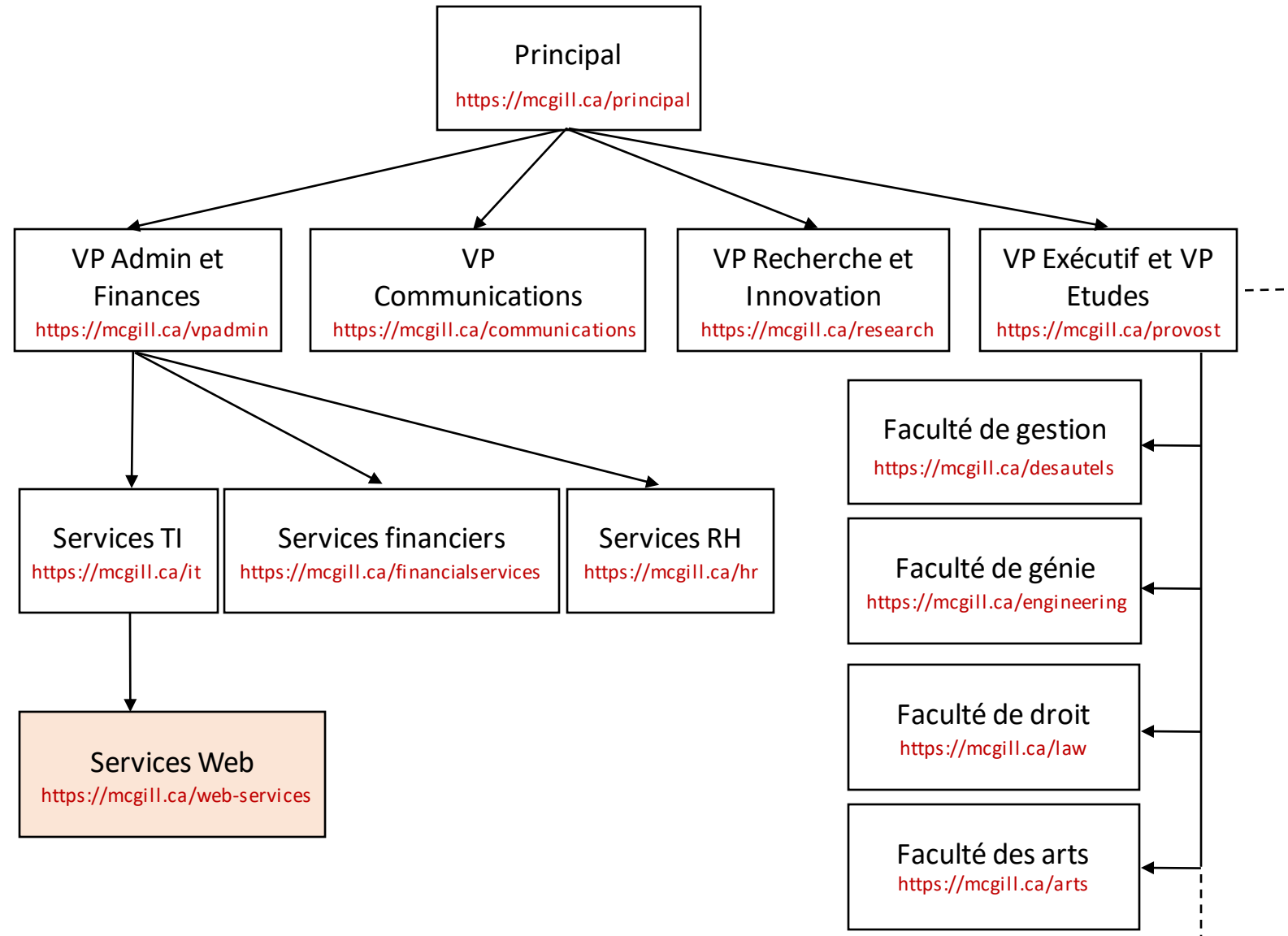


Equipe des Services Web de McGill

Equipe intégrée aux services TI centraux. Offre des sites web uniformes aux différents départements de l'Université.

10 personnes incluant:

- Gestionnaire
- Développeurs backend
- Développeurs frontend
- Analystes en soutien et communication



Exemples de sites web

Facultés

- <https://www.mcgill.ca/arts>
- <https://www.mcgill.ca/music>
- <https://www.mcgill.ca/science>

Administration

- <https://www.mcgill.ca/hr>
- <https://www.mcgill.ca/it>

Vie étudiante

- <https://www.mcgill.ca/campus-life>
- <https://www.mcgill.ca/studentsservices>

Départements d'études

- <https://www.mcgill.ca/surgery>
- <https://www.mcgill.ca/philosophy>
- <https://www.mcgill.ca/geography>

Catalogue de cours et admissions

- <https://www.mcgill.ca/study>
- <https://www.mcgill.ca/admissions>
- <https://www.mcgill.ca/exams>

Services

- <https://www.mcgill.ca/directory>
- <https://www.mcgill.ca/search>

Musées

- <https://www.mcgill.ca/redpath>
- <https://www.mcgill.ca/medicalmuseum>

Relations externes

- <https://www.mcgill.ca/newsroom>

Laboratoires de recherche, conférences, etc...

Exemples de sites web

<https://www.mcgill.ca/science>

The screenshot shows the McGill Faculty of Science website. At the top left is the McGill logo and the text "Faculty of Science". To the right is a search bar and a "Quick Links" dropdown menu. Below the header is a navigation bar with links: About, Undergraduate, Graduate, Research, Outreach, Faculty & Staff, Kudos, Alumni, Supporting Science, and Contact. The main content area features two large images: a student studying at a desk and a scientist using a microscope. Below these images are two columns of links: "Students" with links to Undergraduate, Graduate, and Teaching and learning in the Faculty of Science; and "Research" with links to Research news and events, Undergraduate research opportunities, and Services for Faculty of Science researchers. A "Faculty of Science News" section follows, listing four recent news items with their publication dates and small thumbnail images. At the bottom left is a "VIEW MORE NEWS" link.

<https://www.mcgill.ca/about>

The screenshot shows the McGill About McGill website. At the top left is the McGill logo and the text "About McGill". To the right is a search bar and a "Quick Links" dropdown menu. Below the header is a navigation bar with links: Administration & Governance, History, Notable alumni, and 2022 Quick Facts. The main content area features a large image of a McGill building surrounded by trees with autumn foliage. Below the image is the "Who We Are" section, which includes a paragraph about McGill's status as a leading Canadian research university and a list of five buttons: "Our mission", "In the community", "Our city: Montréal", "Publications", and "Affiliations". The "History" section follows, stating that McGill is a public university founded in 1821. The "A Tradition of Success" section describes McGill's global recognition and lists several notable achievements, including Ernest Rutherford's Nobel Prize-winning research and the invention of the artificial blood cell and Plexiglas.

Infrastructure



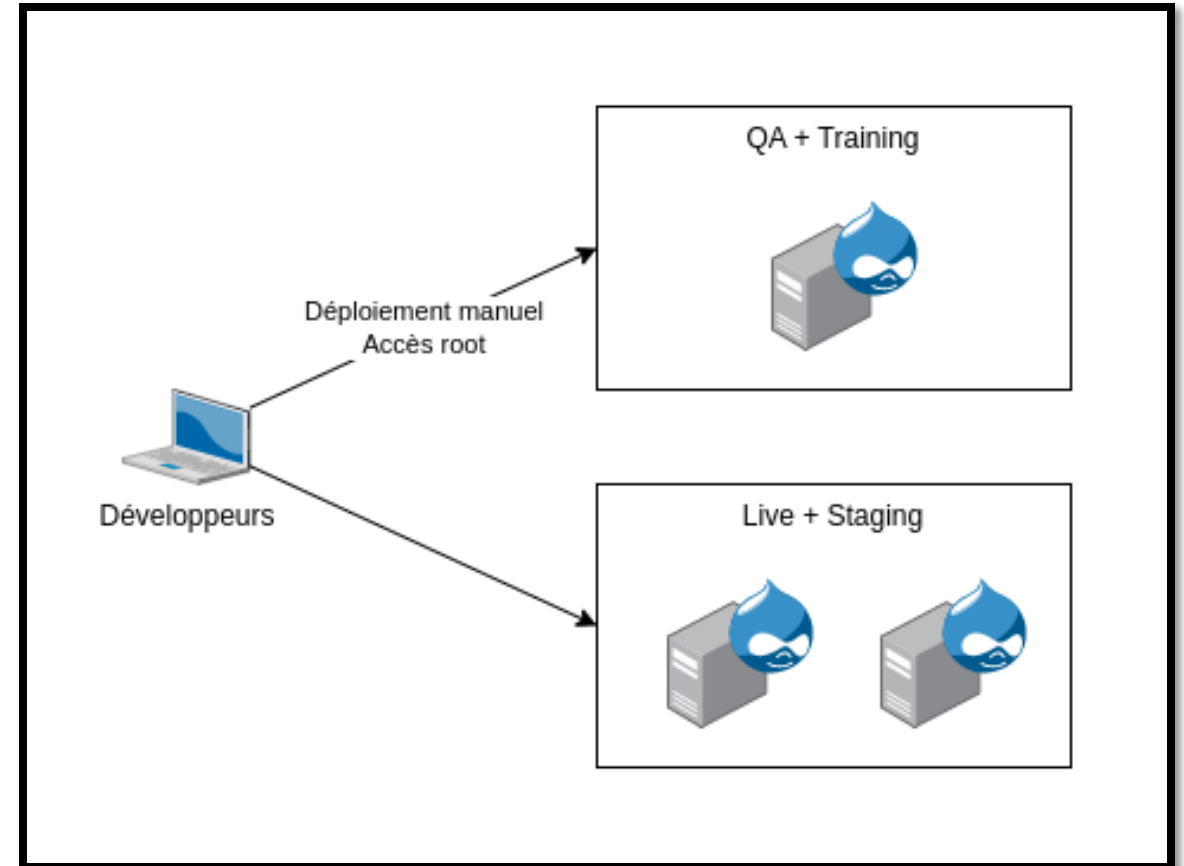
Déploiement (2013)

Déploiement de l'infrastructure:

- Manuel
- Avec un accès root (sudo)
- Sur des machines multi-environnements

Problèmes:

- Divergence entre environnements
- Divergence entre serveurs d'un même environnement
- Permissions inappropriées
- Manque de transparence
- Aucune trace des changements



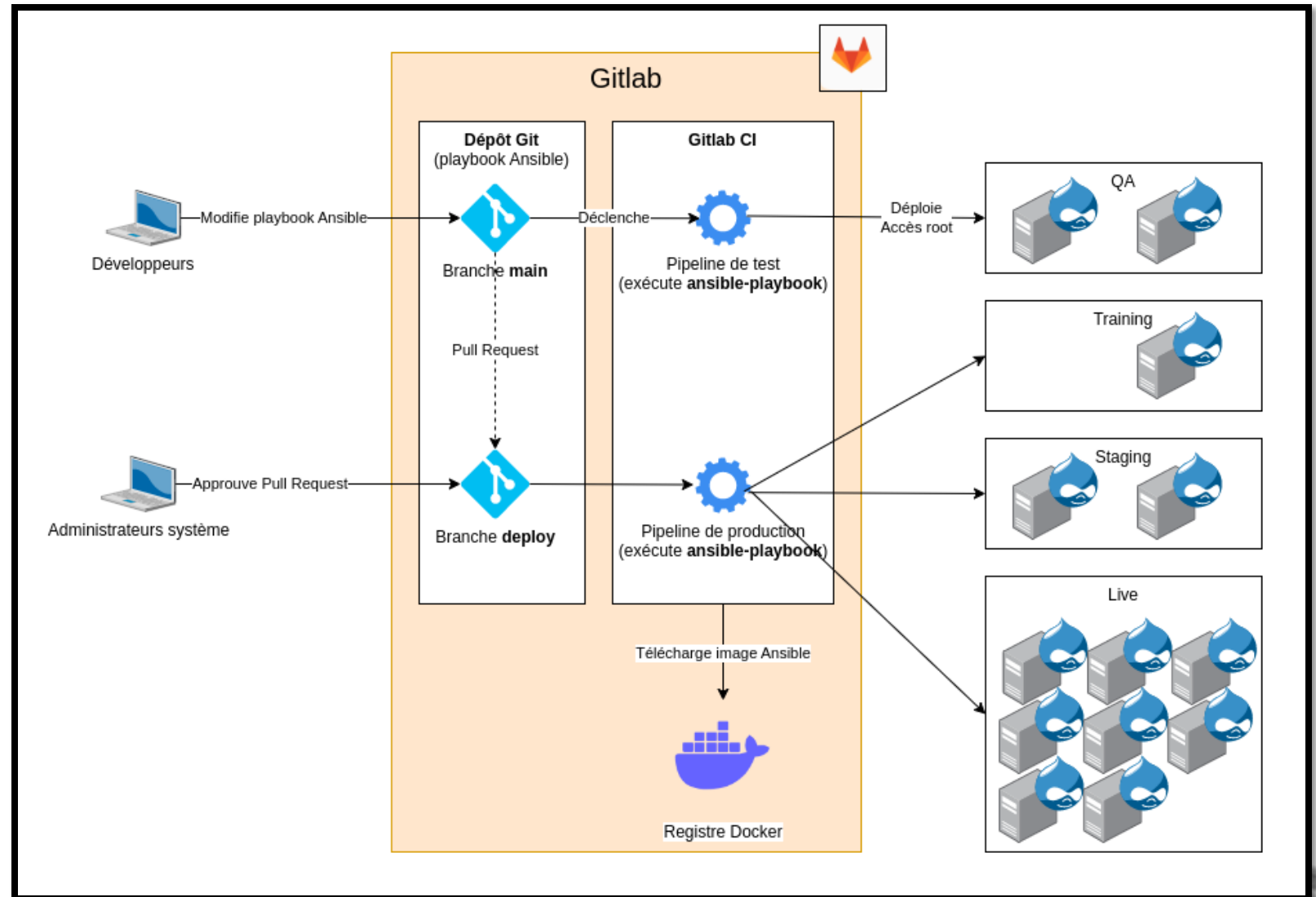
Déploiement (2017)

Déploiement de l'infrastructure:

- Automatisé
- Sans accès root direct
- Sur des VMs dédiées à chaque environnement

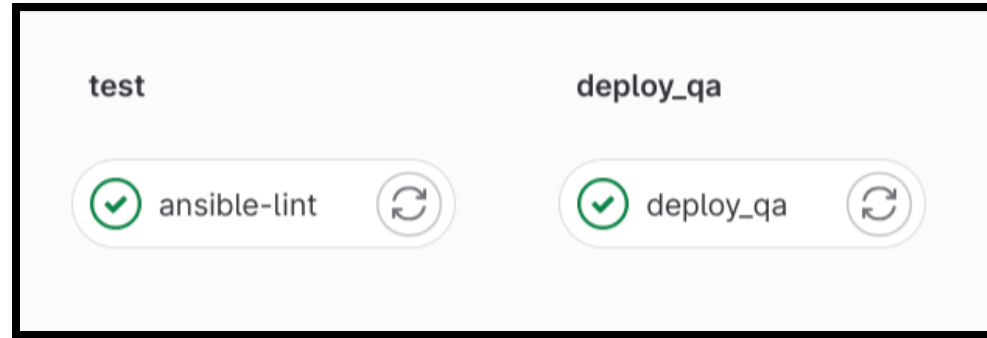
Garantit:

- Similitude entre environnements
- Similitude entre serveurs d'un même environnement
- Permissions limitées, réduit le risque d'erreur humaine
- Transparence avec les autres développeurs et les administrateurs système
- Trace des changements

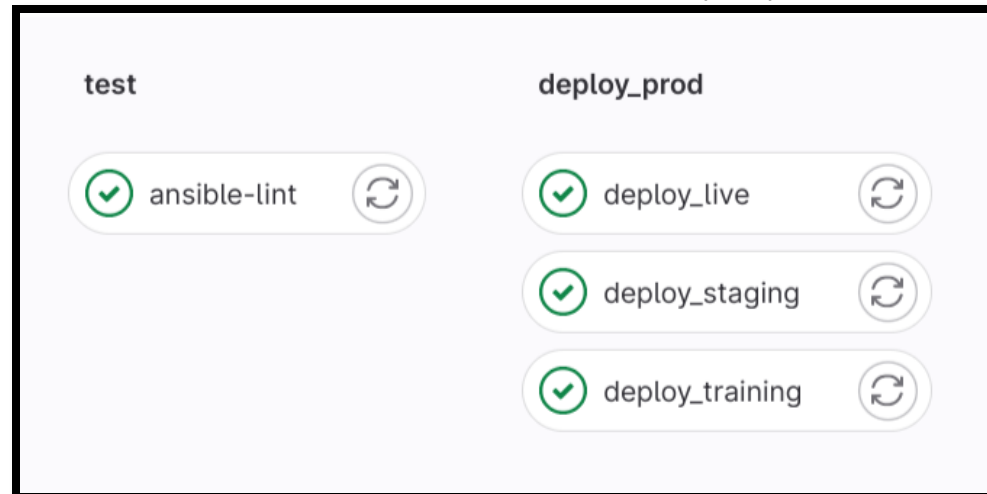


Pipelines Gitlab CI (infrastructure)

Pipeline de test (branche **main**):



Pipeline de production (branche **deploy**):



Jobs Gitlab CI

Logs des changements (i.e. des exécutions du playbook):

The screenshot shows the job log for 'deploy_live'. The log contains the following content:

```
3929 TASK [web : httpd service] *****
3930 ok: [web1.mcgill.ca]
3931 ok: [web2.mcgill.ca]
3932 ok: [web3.mcgill.ca]
3933 ok: [web4.mcgill.ca]
3934 ok: [web5.mcgill.ca]
3935 ok: [web6.mcgill.ca]
3936 ok: [web7.mcgill.ca]
3937 ok: [web8.mcgill.ca]
3938 PLAY RECAP *****
3939 web1.mcgill.ca      : ok=250  changed=18  unreachable=0  failed=0
3940 web2.mcgill.ca      : ok=252  changed=18  unreachable=0  failed=0
3941 web3.mcgill.ca      : ok=252  changed=18  unreachable=0  failed=0
3942 web4.mcgill.ca      : ok=252  changed=18  unreachable=0  failed=0
3943 web5.mcgill.ca      : ok=252  changed=18  unreachable=0  failed=0
3944 web6.mcgill.ca      : ok=252  changed=18  unreachable=0  failed=0
3945 web7.mcgill.ca      : ok=252  changed=18  unreachable=0  failed=0
3946 web8.mcgill.ca      : ok=252  changed=18  unreachable=0  failed=0
3948 Cleaning up project directory and file based variables 00:01
3950 Job succeeded
```

On the right side of the screenshot, the job details for 'deploy_live' are shown:

- Duration: 10 minutes 52 seconds
- Finished: 2 weeks ago
- Queued: 1 second
- Timeout: 1h (from project)
- Runner: #635 (wdtapOGMx)
- Tags: redhat
- Commit: 6f658875
- Merge branch 'main' into 'deploy'
- Pipeline #166836 for deploy
- Current stage: deploy_prod
- Next stages: deploy_live, deploy_staging

Historique des changements:

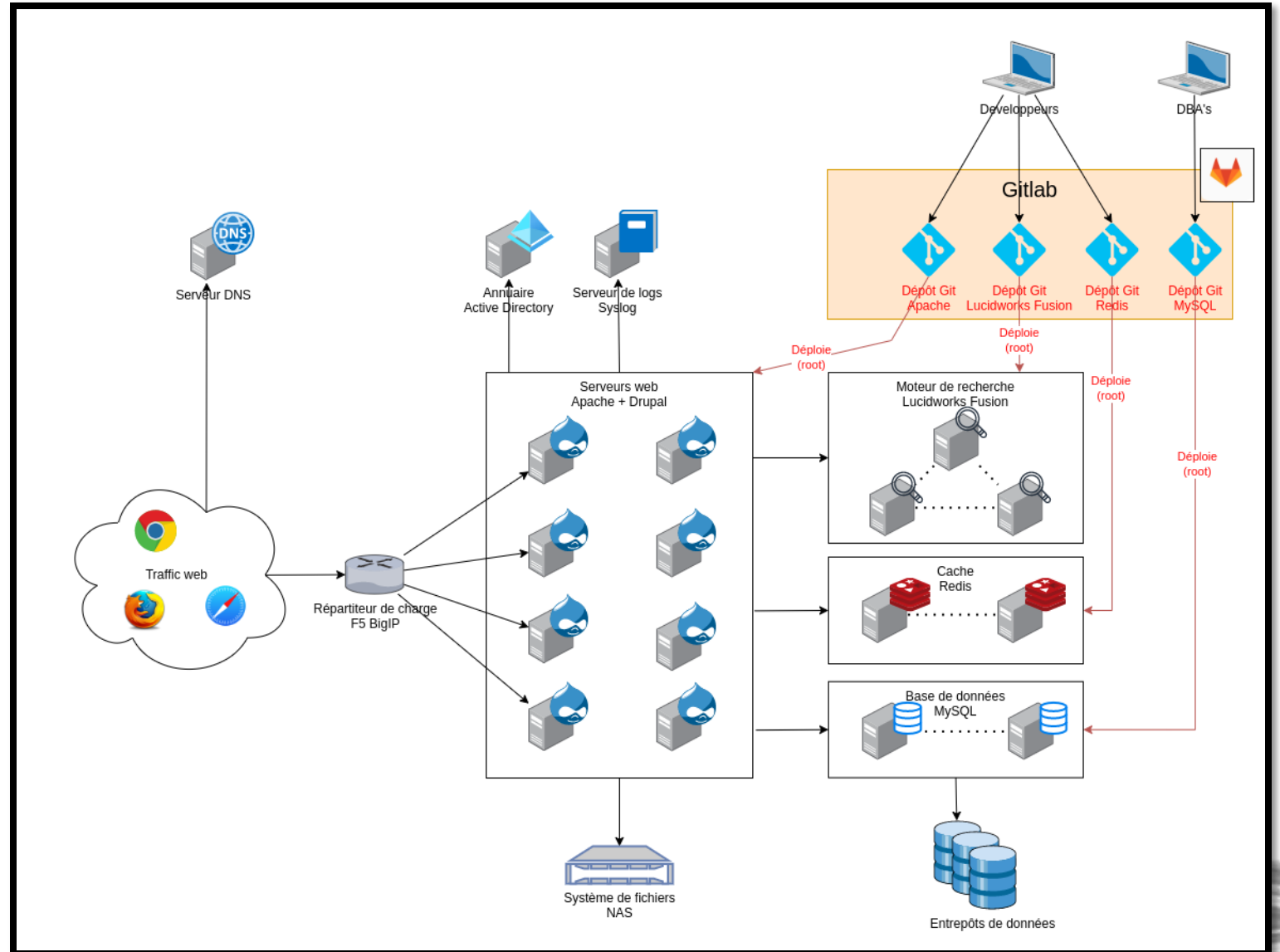
Status	Pipeline	Triggerer	Stages
passed 00:05:01 1 day ago	Merge branch 'WEB-2577-redis-sent...' #171203 main -> 5b2700b4 latest		✓ ✓
passed 00:03:50 4 days ago	Merge branch 'WEB-2522-httpd-sysl...' #170251 main -> ff532283		✓ ✓
passed 00:04:16 1 week ago	Merge branch 'WEB-2322-mysql-swi...' #168275 main -> b889a2df		✓ ✓
passed 00:04:19 2 weeks ago	Merge branch 'WEB-2454-sentinel-s...' #167700 main -> b8a82283		✓ ✓
passed 00:04:17 2 weeks ago	Merge branch 'WEB-2541-csp-heade...' #166419 main -> 6540115b		✓ ✓

Déploiement (2023)

Ansible est devenu notre standard pour déployer les nouvelles infrastructures.

Notre infrastructure peut être découverte en lisant les playbooks Ansible ("Infrastructure as Code").

Nous pouvons comprendre comment sont configurés les serveurs gérés par d'autres équipes (e.g. serveurs MySQL gérés par les DBA's).



Application



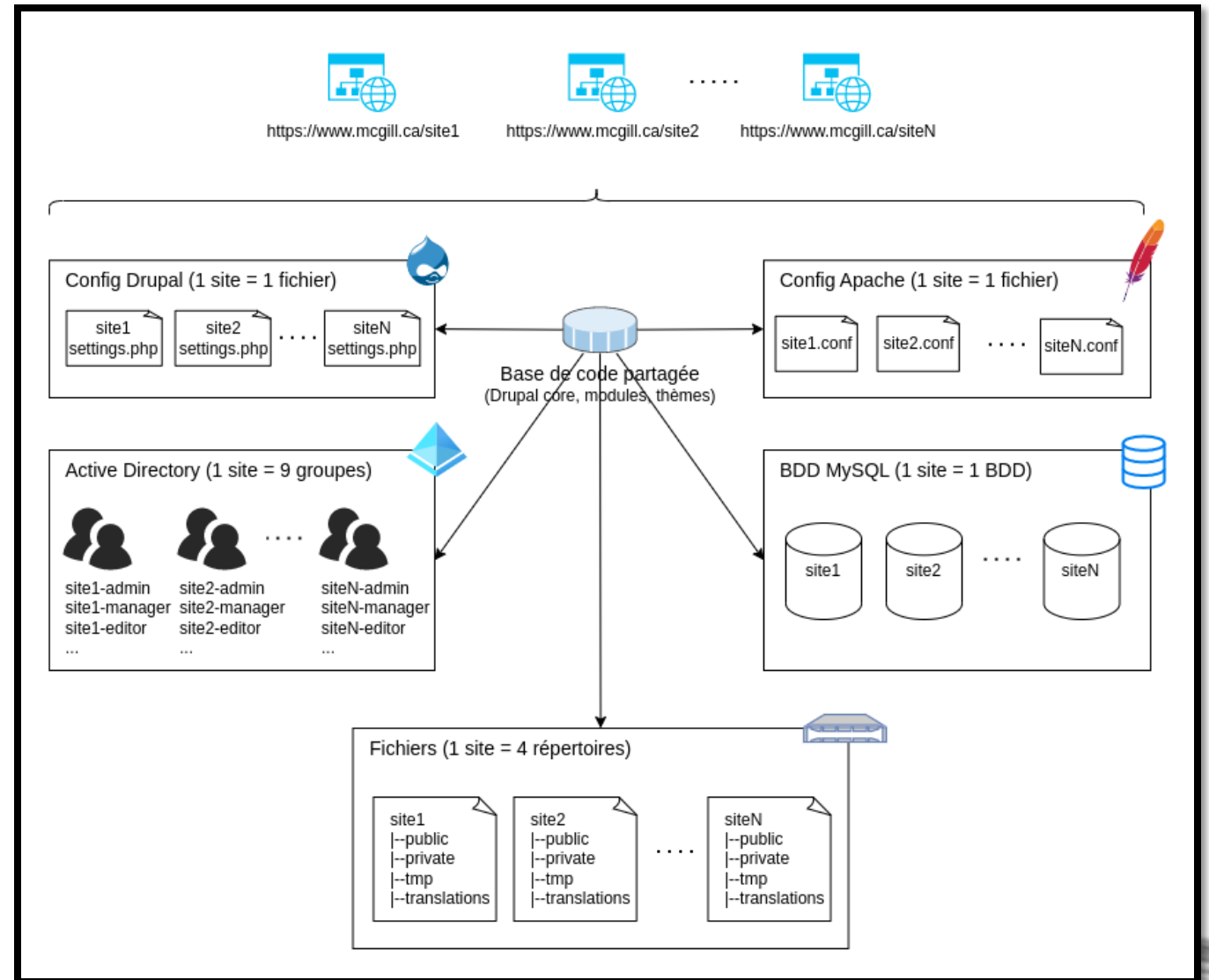
Installation multisite Drupal

1 site =

- 1 fichier de config Apache
- 1 fichier de config Drupal
- 1 base de données MySQL
- 4 répertoires (fichiers utilisateurs)
- 9 groupes Active Directory

Base de code partagée entre tous les sites:

- Drupal et ses dépendances
- Modules Drupal
- Thèmes Drupal



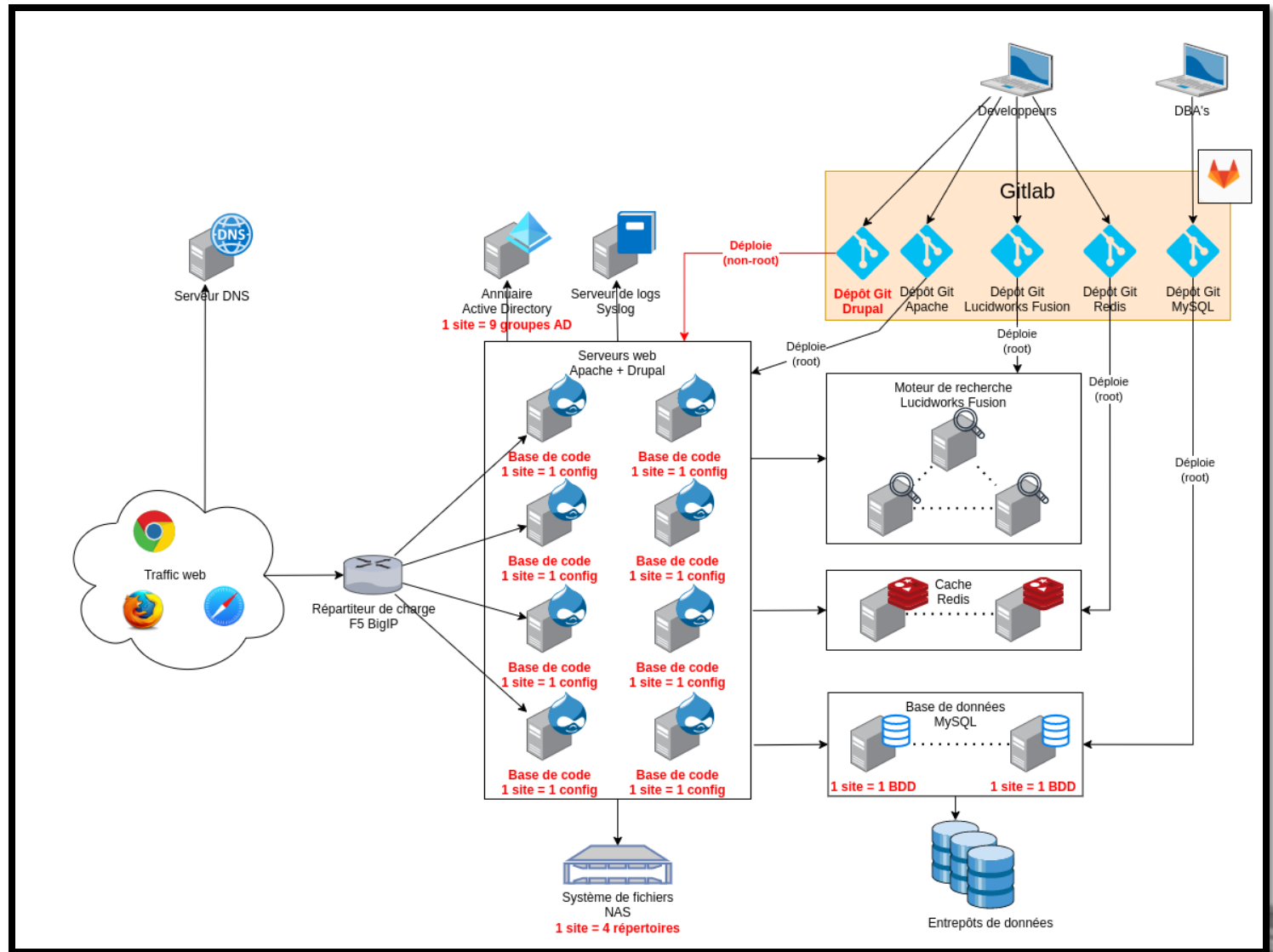
Déploiement de l'application

Même principe que pour l'infrastructure:

- Dépôt Git contient le playbook Ansible
- Pipelines exécutent Ansible (1 job = 1 environnement)

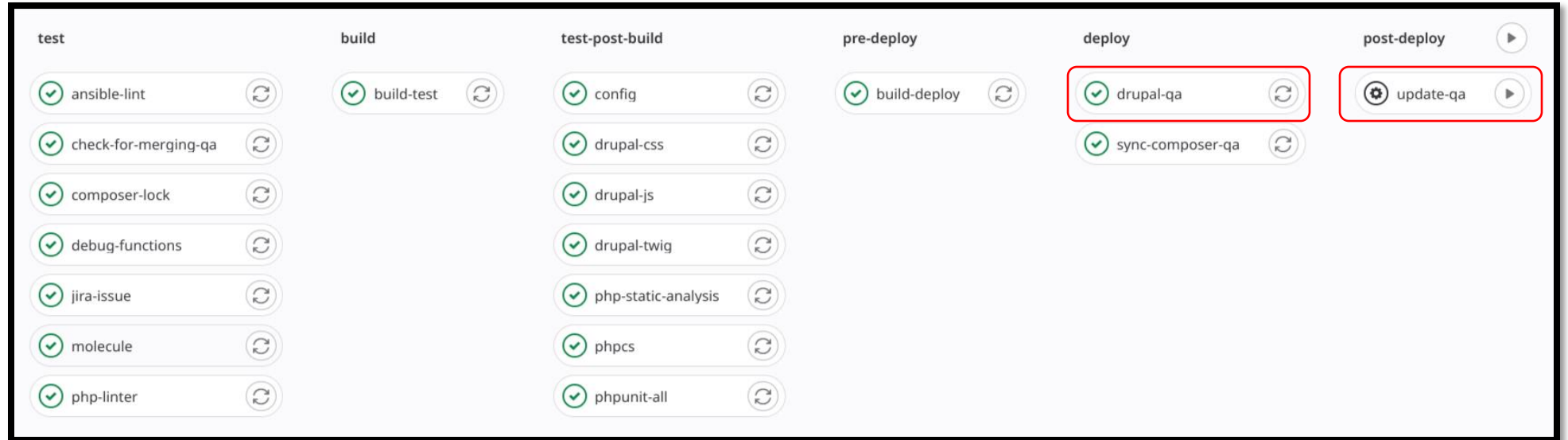
Différences par rapport aux playbooks d'infrastructure:

- Pas d'escalation de privilèges
- Pas d'approbation des Pull Requests par les administrateurs système
- Playbook stocké dans le même dépôt Git que notre code

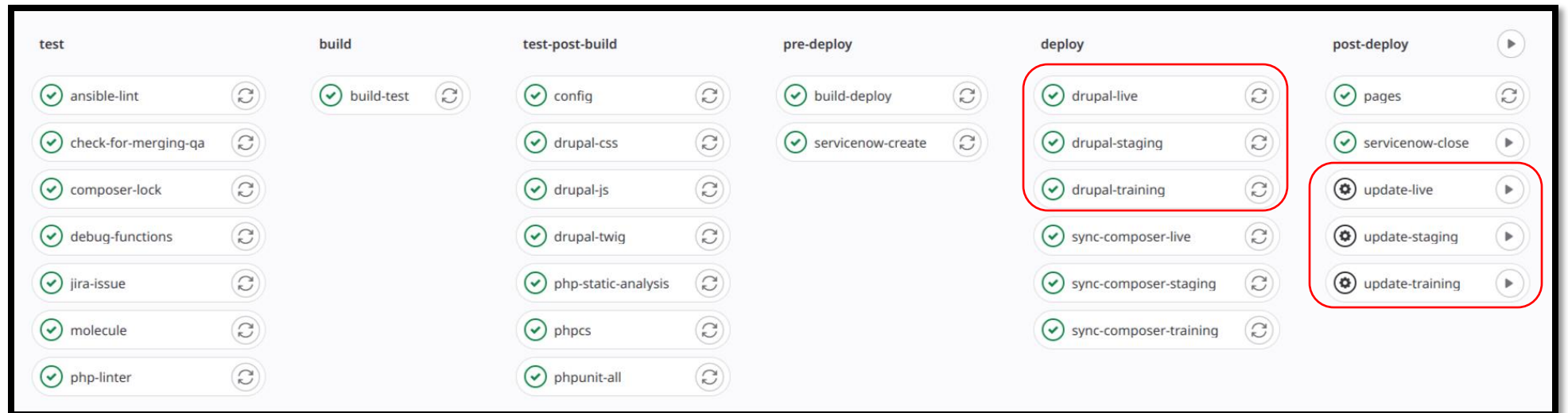


Pipelines Gitlab CI (application)

Pipeline de test
(branche **qa**)



Pipeline de production
(branche **main**)



Pipelines Gitlab CI (application)

Job qui exécute le playbook

```
# .gitlab-ci.yml
drupal-live:
  image: registry.mcgill.ca/cicd/ansible:2.14
  script:
    ...
    - ansible-playbook drupal.yml --diff --limit live
```

Job qui exécute un script de mise à jour

La commande Ansible ad hoc s'occupe de la connection SSH

```
# .gitlab-ci.yml
update-live:
  image: registry.mcgill.ca/cicd/ansible:2.14
  script:
    ...
    - ansible live[0] -m command --args "{{ drupal_root }}/scripts/update-sites.php"
```

Pipeline de production
(branche **main**)

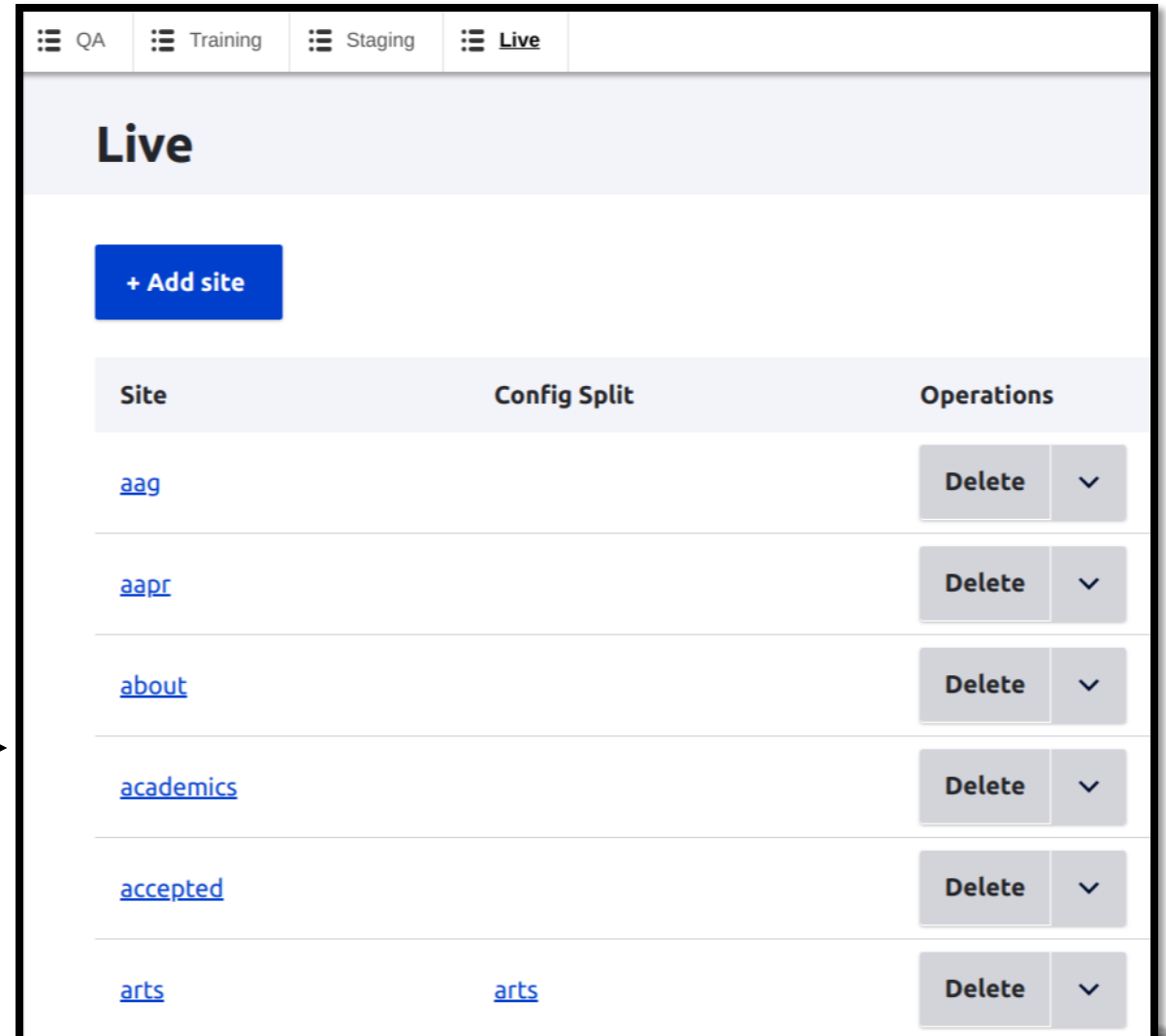
test	build	test-post-build	pre-deploy	deploy	post-deploy
ansible-lint	build-test	config	build-deploy	drupal-live	pages
check-for-merging-qa		drupal-css	servicenow-create	drupal-staging	servicenow-close
composer-lock		drupal-js		drupal-training	update-live
debug-functions		drupal-twig		sync-composer-live	update-staging
jira-issue		php-static-analysis		sync-composer-staging	update-training
molecule		phpcs		sync-composer-training	
php-linter		phpunit-all			

Gestion des sites

Sites listés dans une variable Ansible (source unique de vérité).

Outil interne manipule la variable (fichier YAML du dépôt Git) via l'API Gitlab et déclenche des pipelines.

```
# ansible/group_vars/live/sites
drupal_sites:
  aag: { }
  aapr: { }
  about: { }
  academics: { }
  accepted: { }
  arts:
    config_split: arts
  asap: { }
  ...
```



Site	Config Split	Operations
aag		Delete ▾
aapr		Delete ▾
about		Delete ▾
academics		Delete ▾
accepted		Delete ▾
arts	arts	Delete ▾

Création d'un site

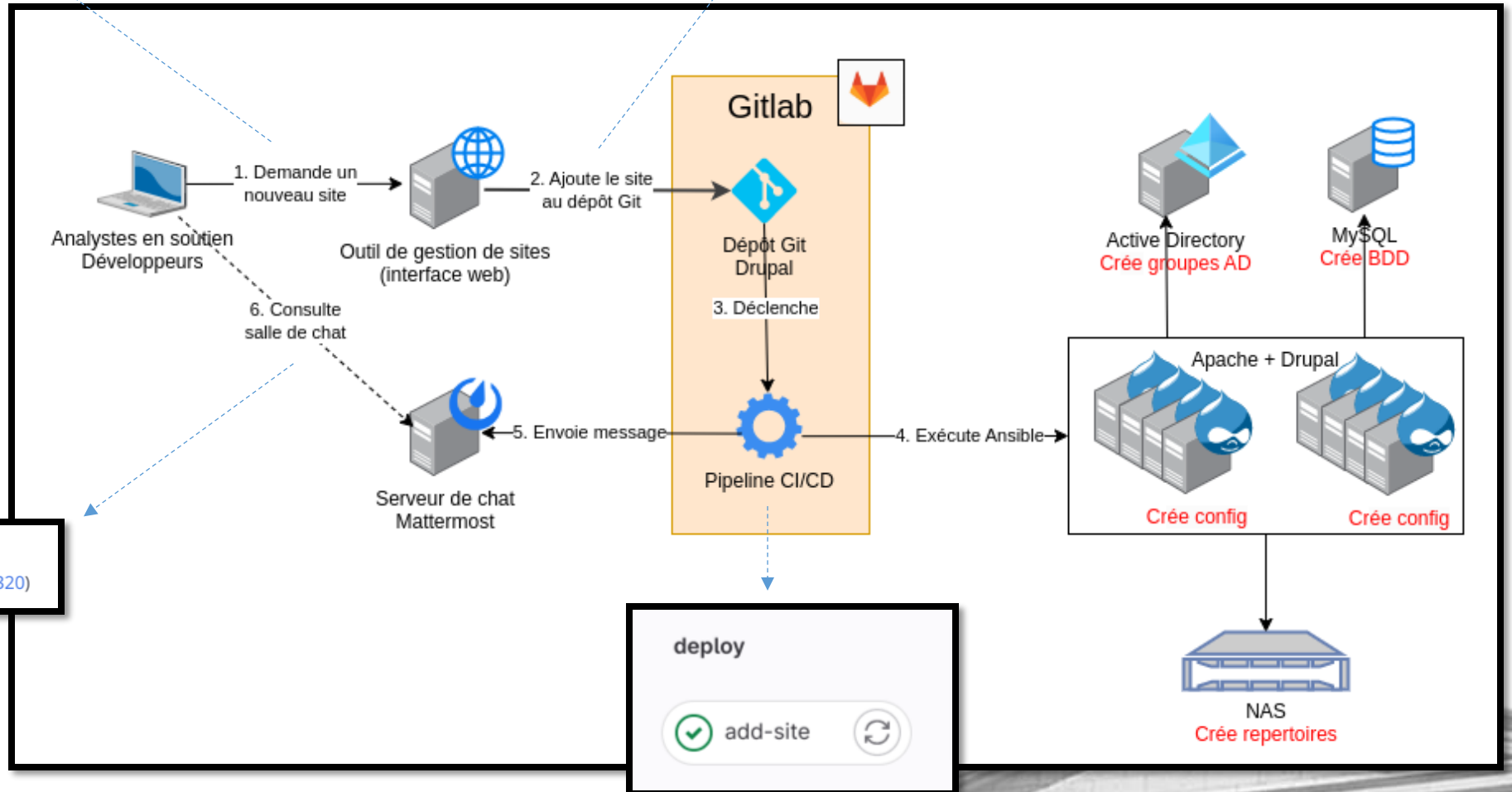
Add new live site

Site*
https://www.mcgill.ca/ new-site

Config Split
- None -

Save

```
# ansible/group_vars/live/sites  
drupal_sites:  
...  
+ neuro: { }  
+ new-site: { }  
newsroom: { }
```



drupal9 BOT 7:03 AM
Created <https://www.mcgill.ca/new-site> (#166320)

deploy

add-site

Développement



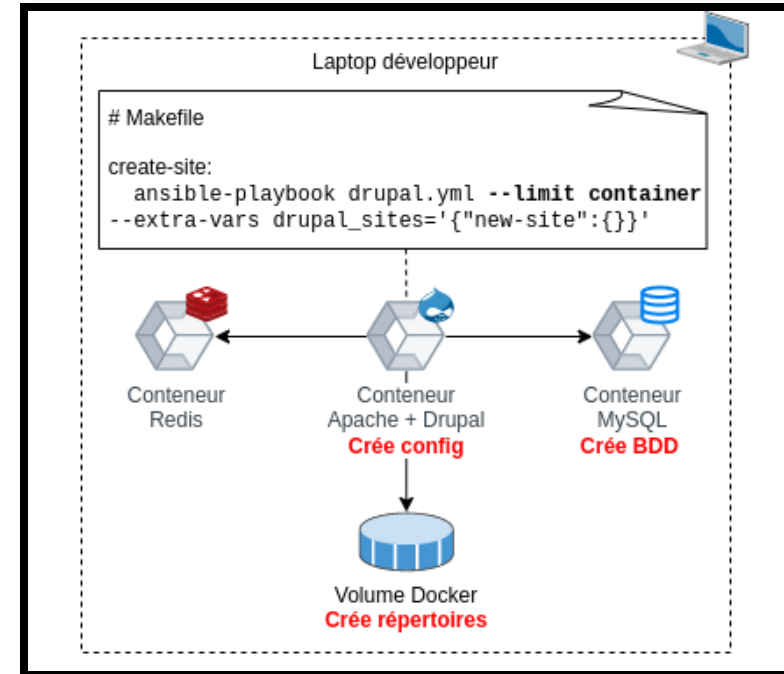
Ansible dans un conteneur

Environnement de développement standardisé avec Docker-Compose.

Réutilisation du playbook applicatif: les sites sont créés en local de la même façon que sur les autres environnements.

Le playbook s'exécute:

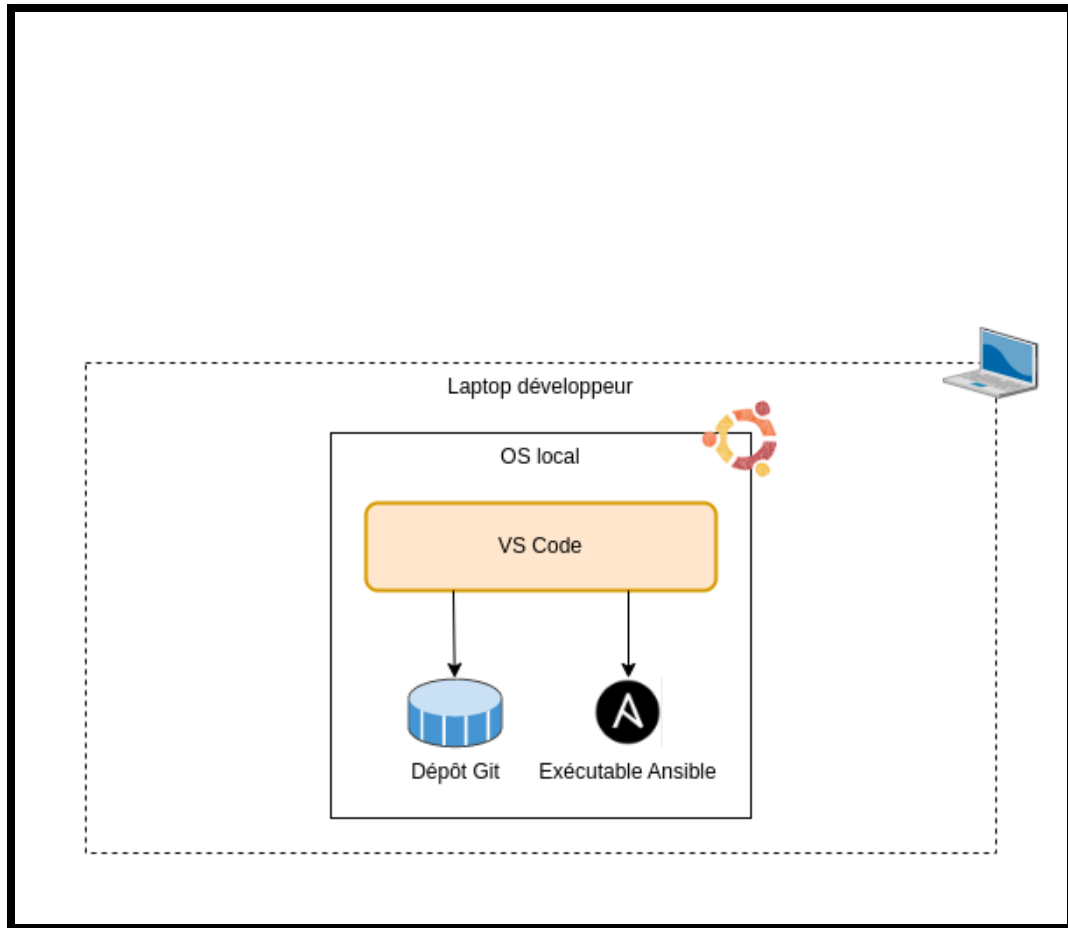
- Avec une connection locale (plutôt que SSH)
- Sur localhost (c'est-à-dire le conteneur) au lieu des serveurs distants



```
# ansible/hosts
container:
  hosts:
    localhost:
      ansible_connection: "local"
qa:
  hosts:
    qa[1:2].mcgill.ca:
      ansible_connection: "ssh"
...
```

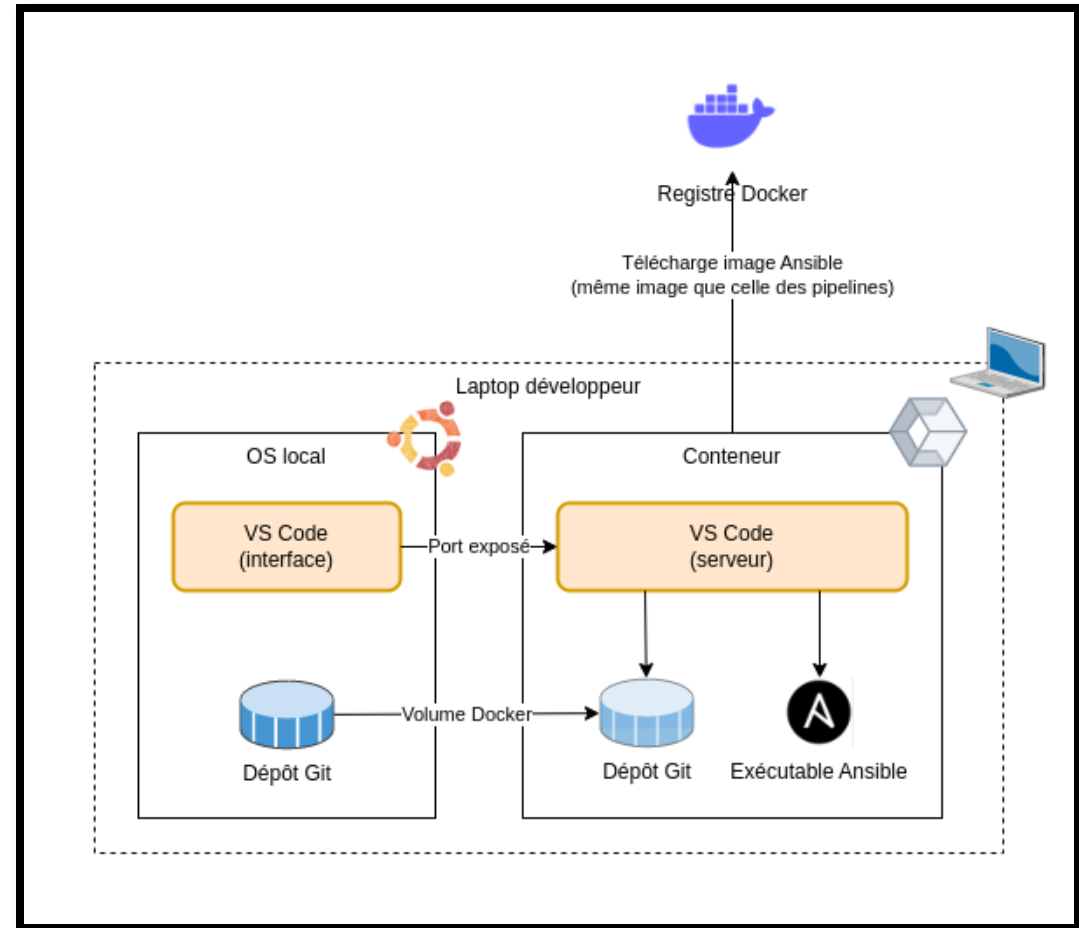
Pipelines: VS Code "Dev Containers"

VS Code classique



VS Code avec l'extension "Dev Containers"

<https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-containers>

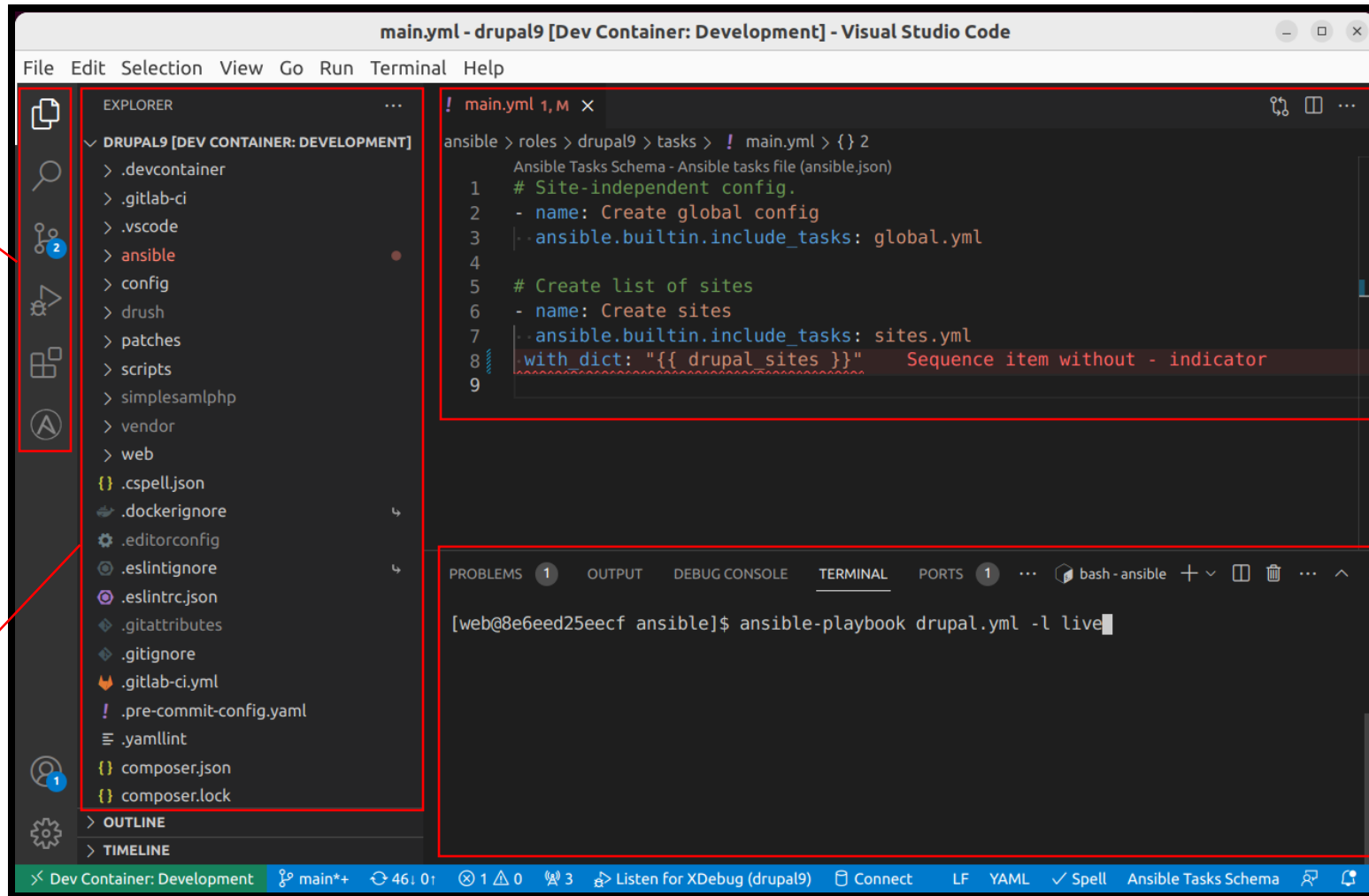


Pipelines: VS Code "Dev Containers"

Développement avec la *même* image Docker que celle des pipelines.

Extensions installées *dans* le conteneur

Navigateur montre les fichiers *dans* le conteneur



Edition des fichiers *dans* le conteneur

Shell s'ouvre *dans* le conteneur

Qualité



Ansible Lint

Revue de code automatisée.

Permet de détecter:

- Erreurs de formatage YAML
- Permissions de fichier risquées
- Modules obsolètes
- Etc

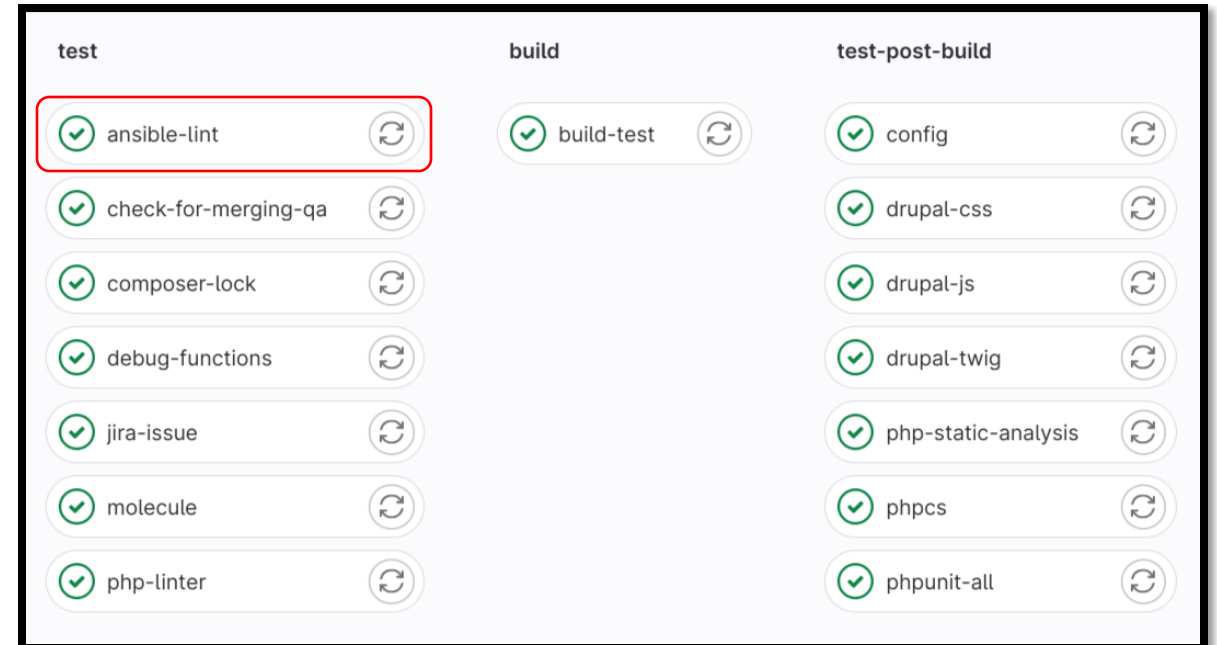
On l'exécute:

1. Dans VS Code
2. Dans un pre-commit hook (Git)
3. Dans la pipeline Gitlab CI

<https://github.com/ansible/ansible-lint>

<https://marketplace.visualstudio.com/items?itemName=redhat.ansible>

<https://pre-commit.com/hooks.html>



```
sh-4.4$ ansible-lint
risky-file-permissions: File permissions unset or incorrect.
roles/drupal/tasks/global.yml:143 Task/Handler: Create sites directory

risky-file-permissions: File permissions unset or incorrect.
roles/drupal/tasks/global.yml:158 Task/Handler: Create services.yml

deprecated-local-action: Do not use 'local_action', use 'delegate_to: localhost'.
roles/drupal/tasks/global.yml:243 Task/Handler: Show result
```

Molecule

Tests automatisés.

Pour chaque rôle Ansible, exécute une séquence de tâches:

1. "Prepare": initialise l'environnement de test
2. "Converge": exécute le rôle Ansible
3. "Verify": teste ce que le rôle a fait
4. "Clean up": nettoie l'environnement de test

Les résultats sont exportés et consultables directement dans l'interface Gitlab, avec nos autres tests automatisés (phpunit).

<https://github.com/ansible-community/molecule>

The screenshot shows three stages of a GitLab CI pipeline:

- test**: ansible-lint, check-for-merging-qa, composer-lock, debug-functions, jira-issue, **molecule** (highlighted), php-linter
- build**: build-test
- test-post-build**: config, drupal-css, drupal-js, drupal-twig, php-static-analysis, phpcs, phpunit-all

Summary

168 tests 0 failures 0 errors 100% success rate 4199.95s



Jobs

Job	Duration	Failed	Errors	Skipped	Passed	Total
phpunit-all	4181.83s	0	0	0	155	155
molecule	18.11s	0	0	0	13	13

Pipelines planifiées

Exécution hebdomadaire des pipelines, sur tous les environnements.

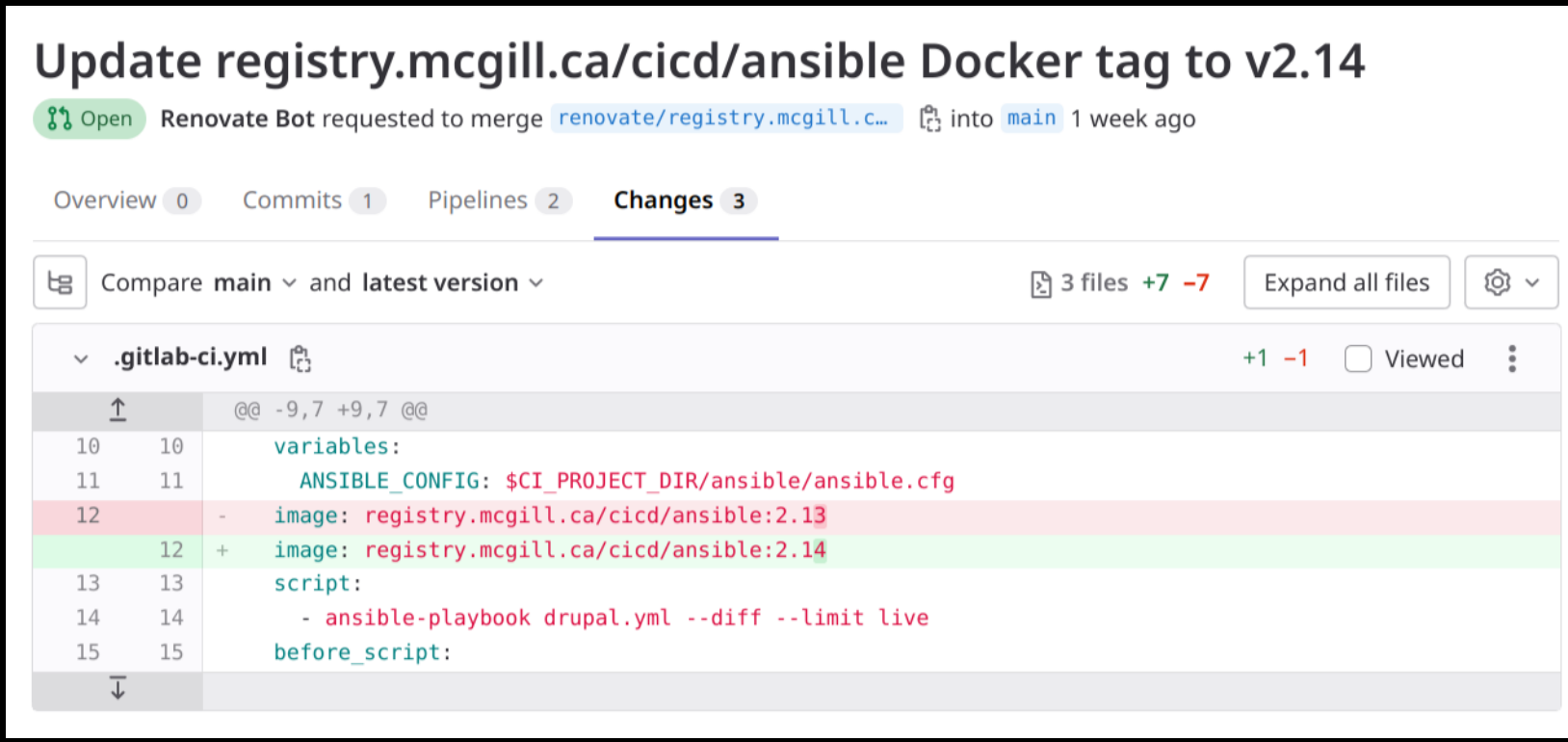
Réconcilie les serveurs avec le playbook Ansible en cas de divergence.

Description	Target	Last Pipeline	Next Run	Owner
Weekly run to prevent Ansible config drift (production)	🏠 main	✅ #167355	in 2 days	
Weekly run to prevent Ansible config drift (qa)	🏠 qa	✅ #166981	in 5 days	

Mises à jour d'Ansible

Pull Requests automatiques lorsqu'une nouvelle version Ansible est disponible.

<https://github.com/renovatebot/renovate>



Update registry.mcgill.ca/cicd/ansible Docker tag to v2.14

Open Renovate Bot requested to merge `renovate/registry.mcgill.c...` into `main` 1 week ago






Overview 0 Commits 1 Pipelines 2 **Changes 3**

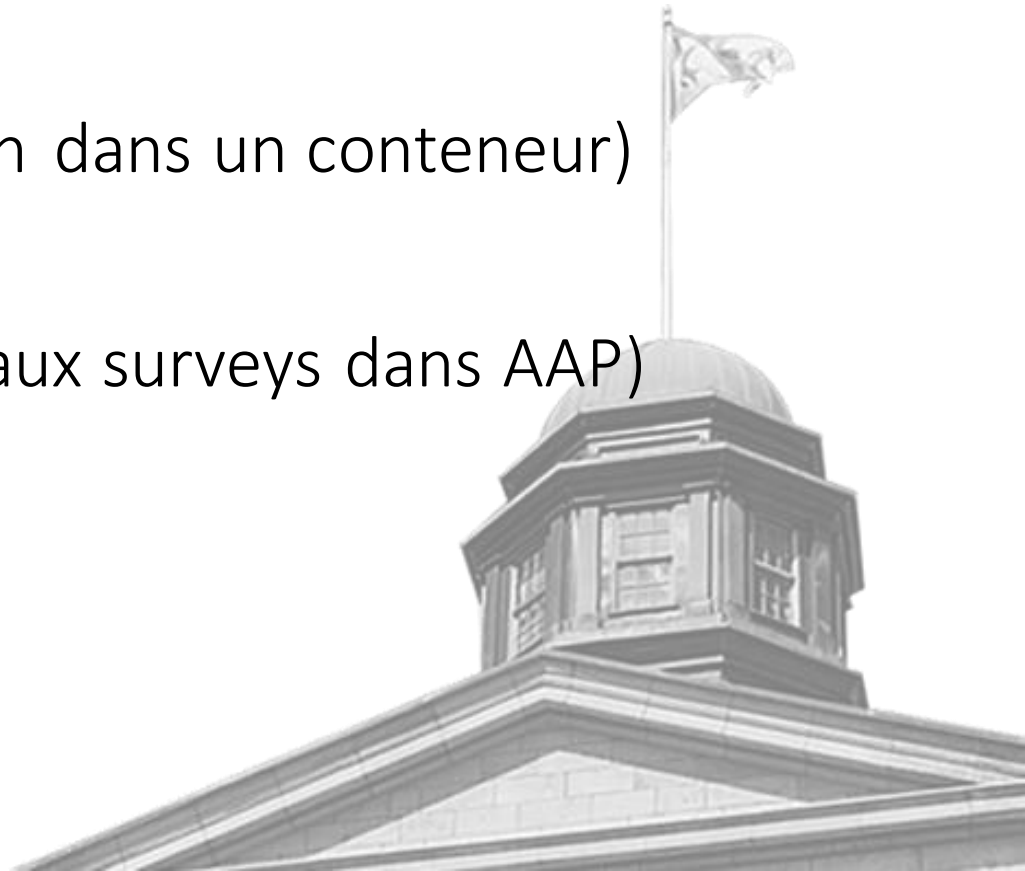
Compare `main` and latest version 3 files +7 -7 Expand all files

`.gitlab-ci.yml` +1 -1 Viewed

```
@@ -9,7 +9,7 @@
10 10     variables:
11 11         ANSIBLE_CONFIG: $CI_PROJECT_DIR/ansible/ansible.cfg
12 -   image: registry.mcgill.ca/cicd/ansible:2.13
12 +   image: registry.mcgill.ca/cicd/ansible:2.14
13 13     script:
14 14         - ansible-playbook drupal.yml --diff --limit live
15 15     before_script:
```

Prochains défis

-  Synchronisation avec notre gestionnaire de secrets (Thycolic)
-  "check mode" (aperçu des changements avant déploiement)
-  ansible-navigator (environnement d'exécution dans un conteneur)
-  Demande de variables au runtime (similaire aux surveys dans AAP)
-  Déploiements dans le Cloud / Kubernetes



Merci

Téléchargez ces diapositives à:
<https://github.com/fengtan/fengtan>

Opportunités de carrière:
<https://www.mcgill.ca/hr/careers>

Thomas Fline



thomas.fline@mcgill.ca



<https://linkedin.com/in/thomasfline>



<https://github.com/fengtan>



<https://drupal.org/u/fengtan>



<https://www.mcgill.ca/it>

Annexes



Optimisation: exécuter Ansible sur un seul site avec jq

Lorsqu'on crée un nouveau site, on n'exécute pas Ansible sur les 1000 sites, cela prendrait trop de temps.

On filtre les sites avec **jq** et on limite l'exécution au seul site qu'on est en train de créer.

<https://jqlang.github.io/jq/>

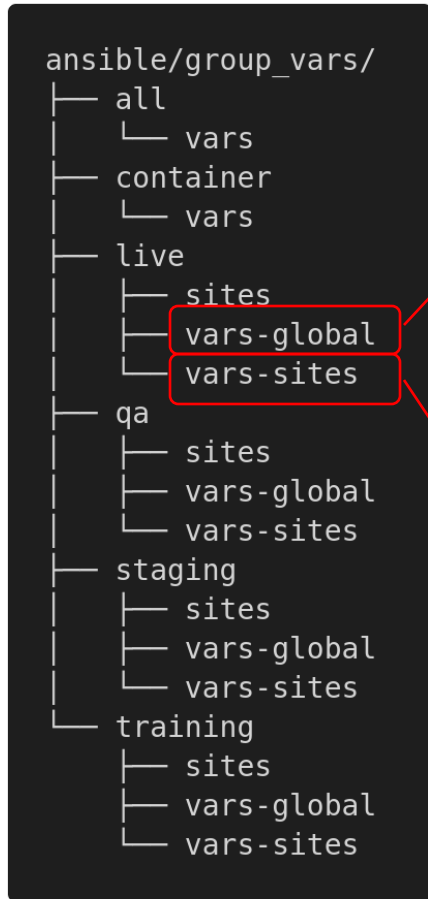
```
# ansible/group_vars/live/sites
drupal_sites:
  aag: { }
  aapr: { }
  about: { }
  academics: { }
  accepted: { }
  arts:
    config_split: arts
  asap: { }
```

```
ansible-playbook drupal.yml --limit live \
  --extra-vars drupal_sites='{"arts":{"config_split":"arts"}}'
```

jq



Optimisation: sélection automatique des tâches avec les tags



```
ansible-playbook drupal.yml --limit live --tags global
```

Déploiement qui change uniquement les variables globales: exécute seules certaines tâches (exécution rapide)

```
ansible-playbook drupal.yml --limit live
```

Déploiement qui change les variables des sites: exécute toutes les tâches (exécution lente: 1000 itérations)

```
# ansible/roles/drupal/tasks/main.yml  
  
# Will run once  
# These tasks use variables in vars-global.  
- name: Global tasks  
  ansible.builtin.include_tasks: global.yml  
  tags: global  
  
# Will run 1000 times (once for each site)  
# These tasks use variables in vars-sites.  
- name: Site creation tasks  
  ansible.builtin.include_tasks: sites.yml  
  with_dict: "{{ drupal_sites }}"
```

