

DAWAN Paris
DAWAN Nantes
DAWAN Lyon

11,rue Antoine Bourdelle, 75015 PARIS
32, Bd Vincent Gâche, 5e étage - 44200 NANTES
Bt Banque Rhône Alpes, 2ème étage - 235 cours Lafayette 69006 LYON



ANSIBLE

Formation Ansible:

Installation, Configuration, Déploiement

Plus d'info sur <http://www.dawan.fr> ou **0810.001.917**

Formateur: Pierre Sablé



Présentation de la formation

Qui suis-je? « mon parcours » « mes compétences »

Exprimer votre besoin:

- Le contexte **IT**
- Votre environnement de travail
- Vos **objectifs** pour cette formation.

Objectifs

Comprendre le mouvement DevOps et l'industrialisation

La philosophie **Ansible**

Installation et configuration de Ansible

Configurations avancées / inventaire / yaml / playbook

Avec ces acquis vous serez en mesure de :

- ✓ Installer et configurer une plateforme Ansible
- ✓ Centraliser l'administration d'un environnement hétérogène
- ✓ Modéliser et déployer une infrastructure type infra-as-code





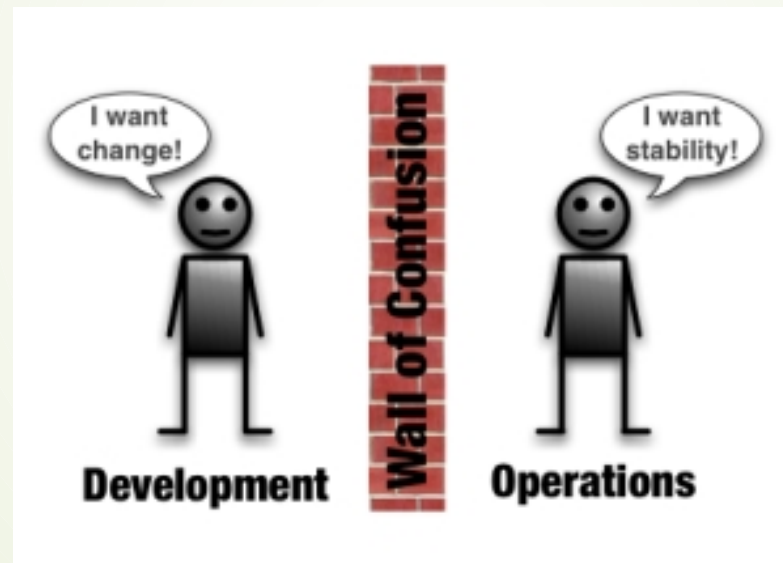
Que vous évoque le terme **DevOps**?

Fonctionnalités d'**Ansible** ?

Quel est le rôle d'un **Playbook**?

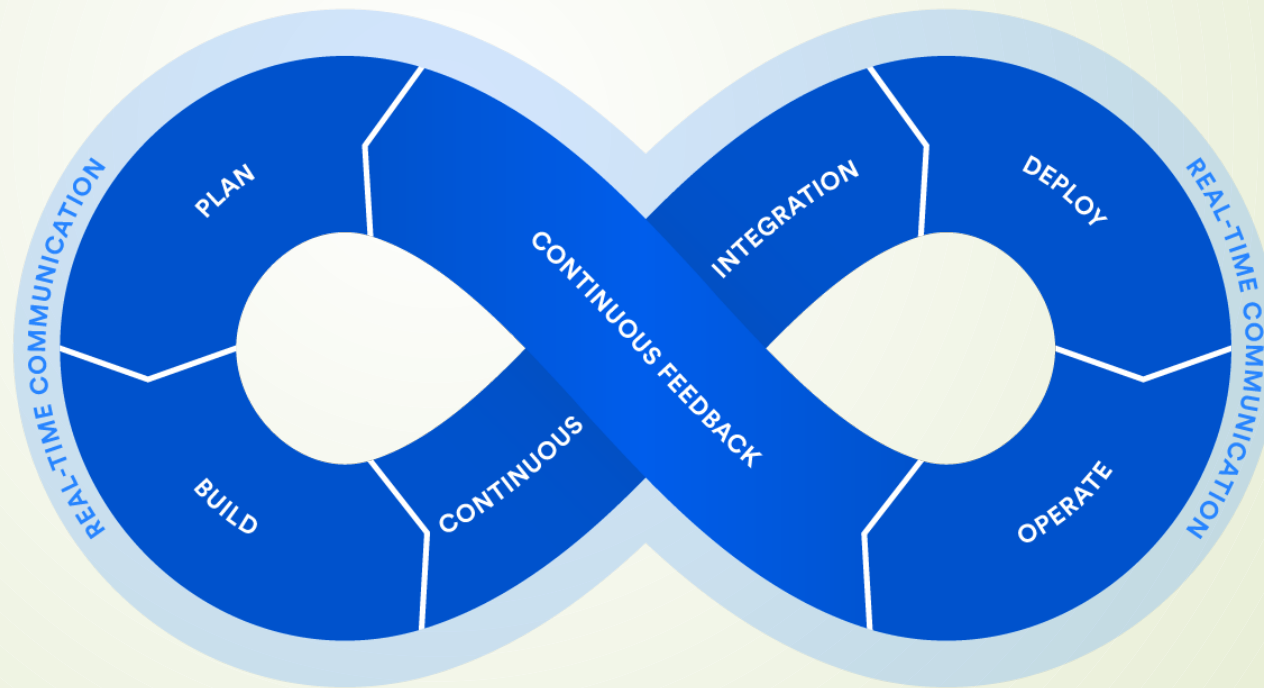
Introduction DevOps

- L'origine du mouvement DevOps



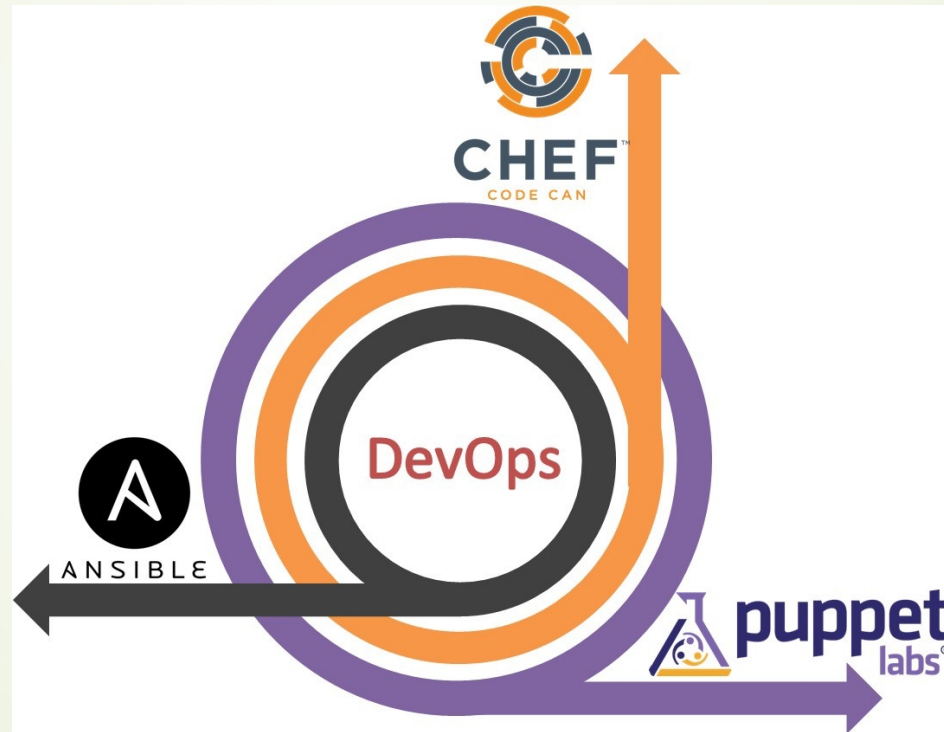
Introduction DevOps

- Les besoins d'industrialisation pour l'exploitation



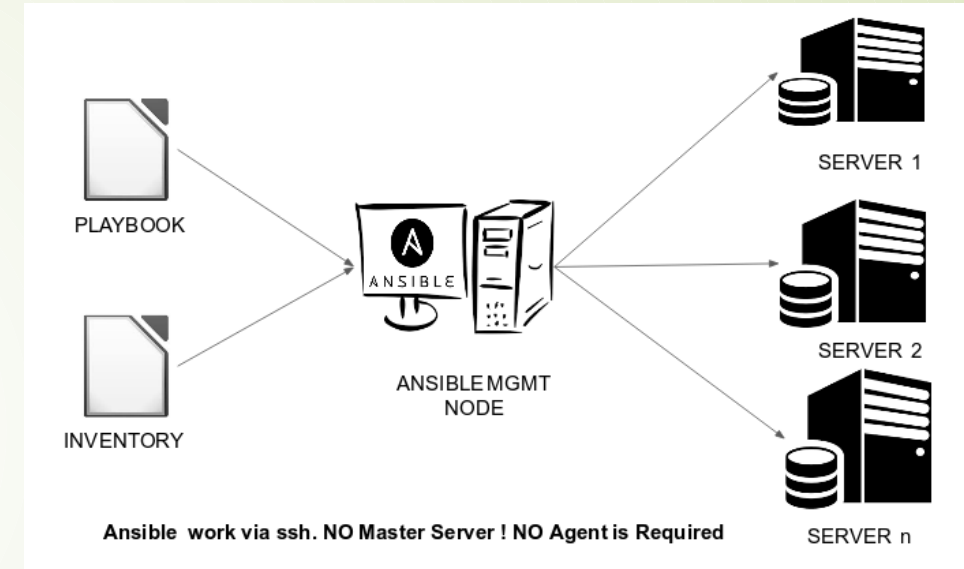
Introduction DevOps

- Les outils



Présentation Ansible

- Origine du projet
- Fonctionnement, idempotence
- Installation
- Machine de contrôle, nœuds, configuration SSH
- L'inventaire, le playbook



Atelier pratique

- Serveur maître
- Les nœuds
- Communication ssh



Le mode « Ad Hoc »

- Format des commandes

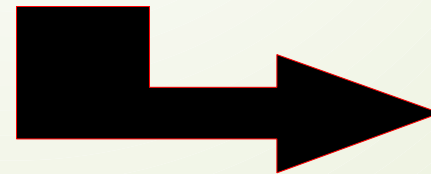
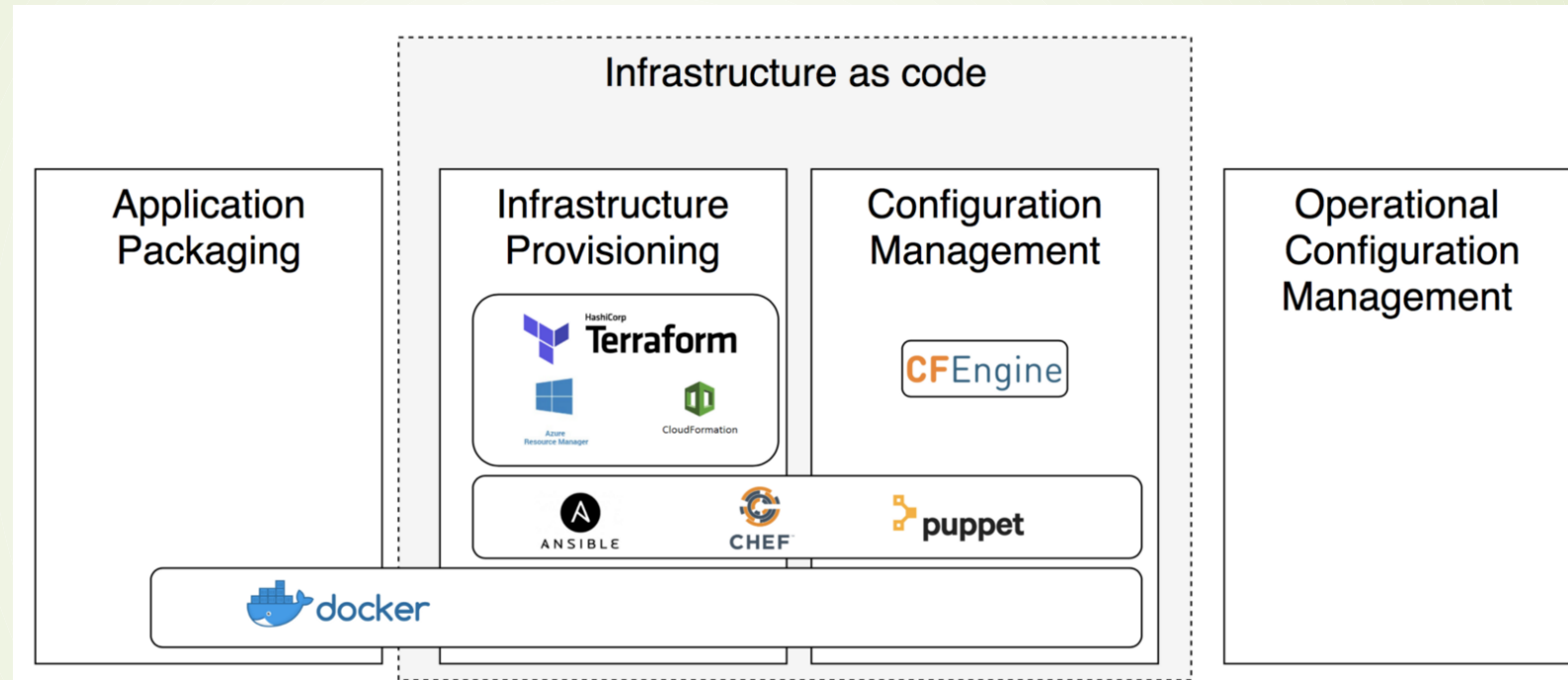
- `$ ansible -i hosts node1 -a 'whoami'`
- `$ ansible -i hosts node2 -m shell -a 'ps -aux | grep ssh'`
- `$ ansible -i hosts -m ping all`

- Modules

`$ ansible -i hosts -m ping all`
⇒ ping, service, yum, copy...

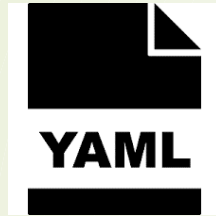


Infrastructure as Code



- yaml
- playbook
- jinja2

Atelier pratique



https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html

- Playbook : configuration users
- Variables et boucles standards
- Passwords + idempotence
- Packages + boucles sur dictionnaire
- Valeurs par défaut
- Secrets : ansible-vault

Atelier pratique



```
{{ ansible_managed }}  
OS: {{ ansible_system }}  
  
{% for element in liste_element %}  
Le contenu de la variable element est le suivant : {{element}}  
{% endfor %}
```


Rôles et structure

- Playbook structuré

```
.
├── hosts                # liste des serveurs
├── playbook.yml         # point d'entrée de votre provisioning
├── roles
│   └── myrole
│       ├── defaults
│       │   └── main.yml # variables par défaut
│       ├── files
│       │   └──          # fichiers statique à uploader
│       ├── handlers
│       │   └── main.yml
│       ├── meta
│       │   └── main.yml # informations meta + dépendances à d'autres rôles
│       ├── tasks
│       │   └── main.yml # tâches (appels de module)
│       ├── templates
│       │   └──          # templates Jinja2
│       ├── tests
│       │   └── test.yml
│       └── vars
│           └── main.yml # variables fortes

```

commande pour créer facilement la structure d'un rôle
\$ ansible-galaxy init myrole



Surcharge des variables

Variable Precedence v2.0

- | | |
|---|--------------------------|
| 1. extra vars | 9. registered vars |
| 2. task vars (only for the task) | 10. host facts |
| 3. block vars (only for tasks in block) | 11. playbook host_vars |
| 4. role and include vars | 12. playbook group_vars |
| 5. play vars_files | 13. inventory host_vars |
| 6. play vars_prompt | 14. inventory group_vars |
| 7. play vars | 15. inventory vars |
| 8. set_facts | 16. role defaults |

- 1. extra vars** : passable à partir de la command line
- 4. role vars** : dans le répertoire /vars d'un rôle
- 8. set_facts** : définie dans les set_facts
- 13. host_vars** : précisée dans les fichiers individuels des machines de l'inventaire
- 14. group_vars** : dans les fichiers de groupes de l'inventaire
- 16. role defaults vars** : la variable par défaut

Boucle jinja2

```
# for.yml
...
vars:
  dict1: {
    'cle1': 'val1',
    'cle2': 'val2'
  }
..
```

```
# templates/for_dict.j2
{% for cle,valeur in dict1.iteritems() %}
  la clé {{ cle }} à pour valeur {{ valeur }}
{% endfor %}
```