



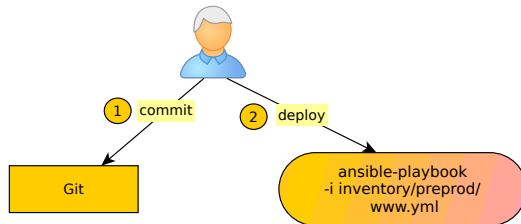
Ansible

gitlab-ci, docker et jmeter

Arnaud Blancher - Ingénieur système @ Orange - baladestrash @ gmail[.]com
Meetup Ansible Paris #12 : Tests, intégration continue et déploiement continue - 07/12/2016

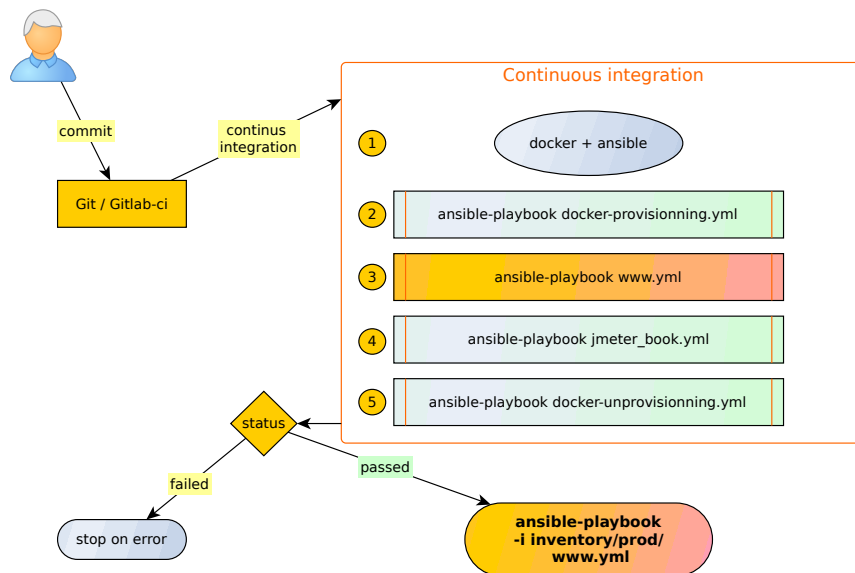
Ansible

sans intégration et déploiement continu



Ansible

avec intégration et déploiement continu



4/40

Use case

2 applications php (app1 et app2) qui répondent chacune à un nom de domaine différent.
Elles sont hébergées sur le même serveur apache

I - Seul au monde

- Ansible
- Jmeter

Ansible

```
|— www.yml
|— roles
|   |— apache2
|   |   |— ...
|   |— app1
|   |   |— ...
|   |— app2
|   |   |— ...
|   |— jmeter
|   |   |— ..
```

```
├── inventory
│   ├── preprod
│   │   ├── 000_hosts
│   │   ├── group_vars
│   │   │   └── all
│   │   │       ├── app1
│   │   │       ├── app1_jmeter
│   │   │       ├── app2
│   │   │       └── app2_jmeter
│   ├── prod
│   │   ├── 000_hosts
│   │   │   └── all
│   │   │       └── ...
│   ├── docker
│   │   ├── 000_hosts
│   │   ├── group_vars
│   │   │   └── all
│   │   │       ├── app1
│   │   │       ├── app1_jmeter
│   │   │       ├── app2
│   │   │       ├── app2_jmeter
│   │   │       └── docker_provisionnement
```

Ansible (v2.1.2)

Déploiement vers les VMs de l'environnement "preprod"

```
ansible-playbook -i inventory/preprod/ www.yml
```

les VMs doivent pré-exister et la clé ssh autorisée.

Jmeter (v2.13)

Objectif : tester le bon fonctionnement de app1 et app2

On souhaite un seul fichier .jmx pour tous les environnements.

Lors du debugage on ne souhaite ni modifier à la main le .jmx pour modifier les variables internes, ni être obligé de fusionner le template après chaque modification.

Jmeter - conception /debug

Lancement local de jmeter avec interface graphique :

```
cd ./roles/app2/templates/etc/jmeter/2.13/scenarii && \  
/usr/local/apache-jmeter-2.13/bin/jmeter \  
-p app2-test-get-slash-prop.properties_en_dur \  
-t app2-test-get-slash-prop.jmx &
```

Jmeter

Variables pré-définies	
Nom : Variables pré-définies	
Commentaires :	
Variables pré-définies	
Nom :	Valeur :
DNS	\${_P(DNS)}
PORT	\${_P(PORT)}

Jmeter à des "variables" \${DNS} et des "propriétés" \${_P(DNS)}.
Seules les propriétés peuvent être passées en ligne de commandes ou via un fichier.

Jmeter

Test sur / du nom de domaine

Requête HTTP		
Nom : <input type="text" value="Get /"/>		
Commentaires :		
Serveur web		
Nom ou adresse IP : <input style="width: 400px;" type="text" value="\${DNS}"/>		Port : <input style="width: 100px;" type="text" value="\${PORT}"/>
Requête HTTP		
Implémentation : <input style="width: 50px;" type="text" value=""/>	Protocole [http] : <input style="width: 50px;" type="text" value=""/>	Méthode : <input style="width: 50px;" type="text" value="GET"/>
Chemin : <input style="width: 150px;" type="text" value="/"/>		

Jmeter

Recherche de la chaîne "APP2" dans la page

Assertion Réponse	
Nom : Assertion Réponse	
Commentaires :	
Appliquer sur	
<input type="radio"/> L'échantillon et ses ressources liées <input checked="" type="radio"/> L'échantillon <input type="radio"/> Les ressources liées <input type="radio"/> Une variable :	
Section de réponse à tester	
<input type="radio"/> Texte de réponse <input checked="" type="radio"/> Document (texte) <input type="radio"/> URL Echantillon <input type="radio"/> Code de réponse <input type="radio"/> Message de réponse <input type="radio"/> Entêtes de réponse <input type="checkbox"/> Ignorer le s	
Type de correspondance du motif	
<input type="radio"/> Contient (exp. régulière) <input type="radio"/> Correspond à (exp. régulière) <input type="radio"/> Est égale à (texte brut) <input checked="" type="radio"/> Contient (texte brut) <input type="checkbox"/> Inverser	
Motifs à tester	
Motifs à tester	
APP2	

Jmeter

Contenu retourné par le serveur

Arbre de résultats

Nom : View Results Tree

Commentaires :

Écrire les résultats dans un fichier ou lire les résultats depuis un fichier CSV / JTL

Nom du fichier :

Texte brut

Résultat de l'échantillonRequêteDonnées de réponse

Get /

APP2 app2.tp3.dev-arnaud.toshiba.local 10.0.0.14

Jmeter

Chez moi ça marche ;-)

II - Systématisons

- la création de conteneur de test
- l'exécution de tests Jmeter

Docker (v1.12)

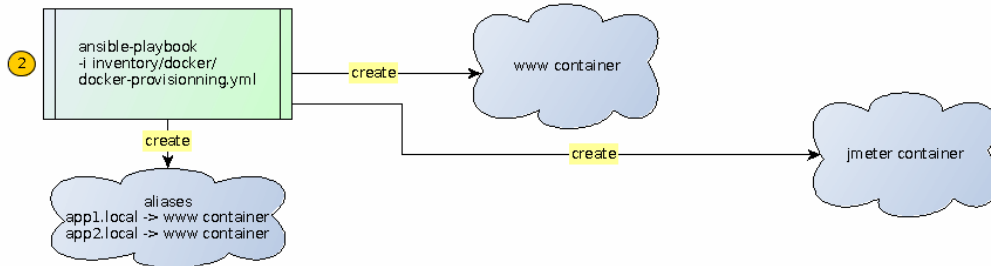
Objectif : créer un environnement vierge et jettable pour tester

Docker pour simuler des VMs en utilisant des conteneurs à la place.
Gain de temps et gestion mémoire plus facile.

Docker

Ansible, créé moi les conteneurs à partir de mon inventory ;-)

```
ansible-playbook -i inventory/docker/ docker-provisionning.yml
```



Docker

inventory/docker/000_hosts :

```
[www]
ansible_tp_www01

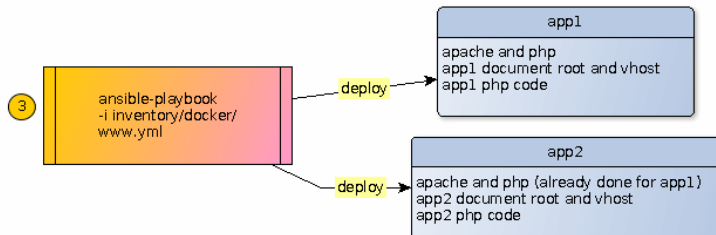
[jmeter]
ansible_tp_jmeter01 docker_provisionning_image="jmeter2.13"

[all:vars]
ansible_connection=docker
```

alias dns pour les conteneurs docker
inventory/docker/group_vars/all/docker_provisionning :

```
docker_provisionning_aliases:
  ansible_tp_www01:
    - "app1.local"
    - "app2.local"
```

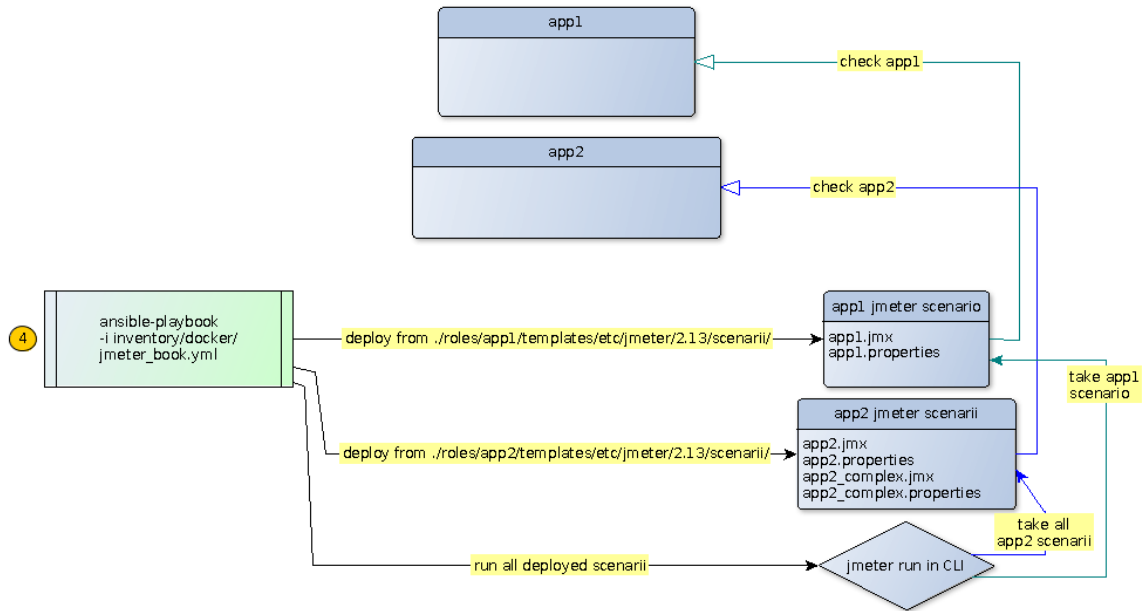
Ansible Déployer vers les conteneurs



Ansible Jmeter

Ansible, transfère les .jmx de toutes mes apps vers le conteneur jmeter et lance les tests ;-)

```
ansible-playbook -i inventory-docker jmeter_book.yml \
    --tags=jmeter_scenarii_copy,jmeter_run_tests
```



Ansible Jmeter

Les .jmx d'une application sont dans un sous répertoire de l'application et déclarés dans la configuration

```
# inventory/docker/group_vars/all/app1_jmeter :
```

```
jmeter_jmx_present:
```

```
"roles/app1/templates/{{jmeter_scenarii_dir}}/app1-test-get-slash.jmx": "app1-test-get-slash.jmx"
```

```
jmeter_jmx_absent:
```

```
"{{jmeter_scenarii_dir}}/app1-test-get-slash-FAKE.jmx": "0"
```

```
# inventory/docker/group_vars/all/app2_jmeter :
```

```
jmeter_jmx_present:
```

```
"roles/app2/templates/{{jmeter_scenarii_dir}}/app2-test-get-slash-prop.jmx": "app2-test-get-slash-prop.jmx"
```

```
jmeter_jmx_absent:
```

```
"{{jmeter_scenarii_dir}}/app2-test-get-slash-FAKE.jmx": "0"
```

Ansible Jmeter

La fusion des hashes est nécessaire dans la configuration d'ansible pour le rôle jmeter

```
# ansible.cfg
[defaults]
hash_behaviour = merge
```


III - Automatisons

- les tests à chaque commit
- le déploiement

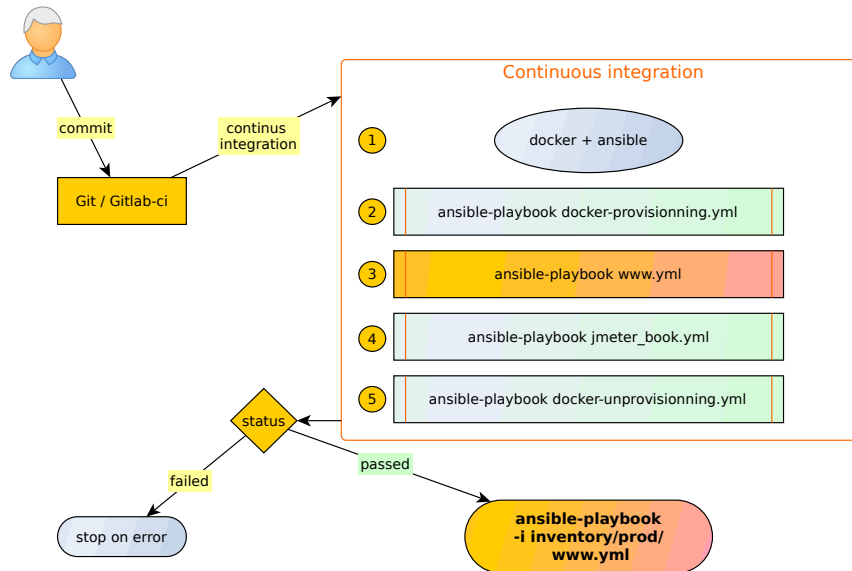
Gitlab

- concurrent de Github
- interface graphique au dessus de git
- gestion des groupes de projet et des utilisateurs
- la version "Community Edition" est open source
- projet très actif

Gitlab-ci

- outil d'intégration continue
- description des étapes de la ci en yaml
- permet de tester vers des VMs, vers des conteneurs docker ou en local
- fichier .gitlab-ci.yml dans le dépôt git
- les actions spécifiques sont appelées via du shell

Gitlab-ci



Gitlab-ci : intégration

- le 'runner' doit être privilégié pour faire du DinD

```
/etc/gitlab-runner/config.toml
[[runners]]
...
privileged = true
```

- utilisation d'une image docker avec ansible et docker

1

docker + ansible

```

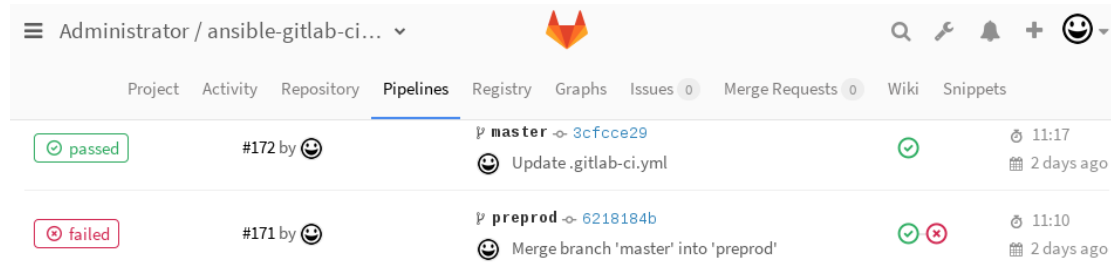
image: arnaudblancher/docker-ubuntu14-ansible-docker
# ordre des stages (construction de conteneur docker à chaque fois)
stages:
- test
- deploy

job_test:
  stage: test
  script:
    # Démarrer docker qui est déjà dans l'image
    - service docker start
    - for i in $(seq 60); do docker info && break || sleep 2; done
    # provisionner les conteneurs dockers correspondant à l'inventaire
    - sudo ansible-playbook --user=root -i ./inventory/docker/ ./docker-provisionning.yml

    # déployer l'ensemble de la plateforme sur les conteneurs docker
    - ansible-playbook --user=root -i ./inventory/docker/ prerequisites.yml
    - ansible-playbook --user=root -i ./inventory/docker/ www.yml
    # tester avec jmeter
    - ansible-playbook --user=root -i ./inventory/docker/ jmeter_book.yml --tags=jmeter_scenarii_copy,:













```

Gitlab-ci



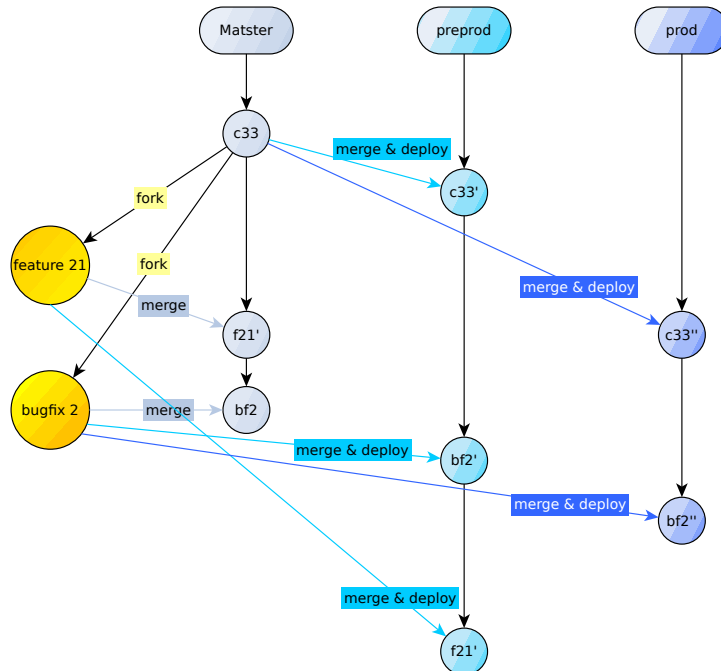
Administrator / ansible-gitlab-ci... ▾

Project Activity Repository Pipelines Registry Graphs Issues 0 Merge Requests 0 Wiki Snippets

✓ passed	#172 by 	 master  3cfcce29  Update .gitlab-ci.yml	✓	 11:17  2 days ago
✗ failed	#171 by 	 preprod  6218184b  Merge branch 'master' into 'preprod'	✓ ✗	 11:10  2 days ago

32/40

Workflow Git



33/40

Gitlab-ci : Déploiement

- le conteneur doit avoir la clé privée pour se connecter aux VMs
- la clé `ANSIBLE_DEPLOY_PRIVATE_KEY` est stockée dans le projet gitlab (et non dans le dépôt git)

Gitlab-ci : Déploiement

```
job_deploy_preprod:
  stage: deploy
  # s'applique seulement à la branche git 'preprod'
  only: preprod
  # est exécuté uniquement lorsque le stage précédent réussit
  when: on_success
  script:

    # Ajout de la clé privée
    - 'which ssh-agent || ( apt-get update -y && apt-get install openssh-client -y )'
    - eval $(ssh-agent -s)
    - ssh-add <(echo "$ANSIBLE_DEPLOY_PRIVATE_KEY")
    - mkdir -p ~/.ssh
    - '[[ -f /.dockerenv ]] && echo -e "Host *\n\tStrictHostKeyChecking no\n\n" > ~/.ssh/config'

    # déployer vers la preprod
    - ansible-playbook --user=root -i ./inventory/preprod/ www.yml
```

Gitlab-ci : Déploiement

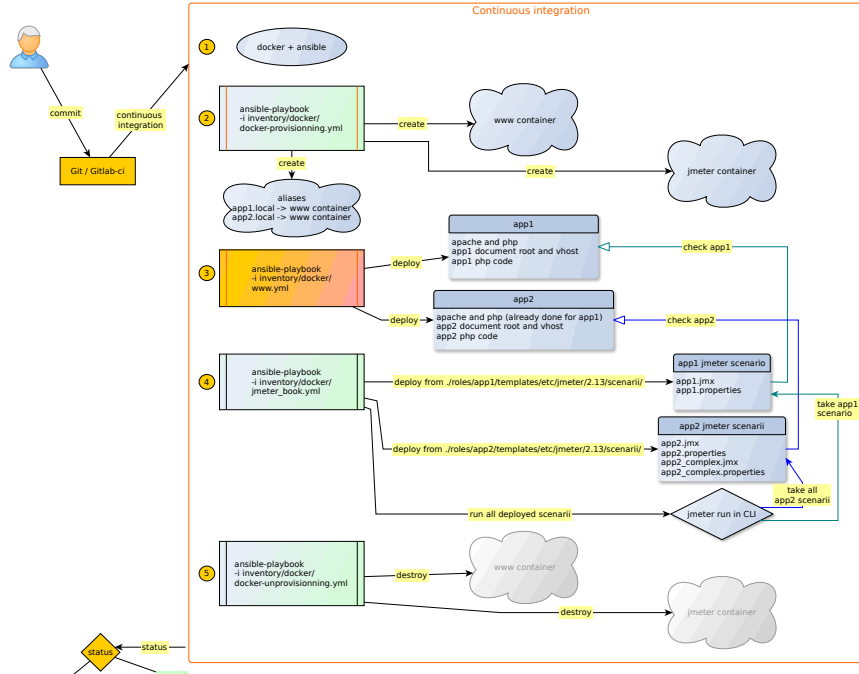
Merge branch 'master' into 'preprod'

wip

See merge request !6



Commit -> CI -> CD



37/40

Limitations

- dépôts distants
- images distantes -> registry interne de gitlab
- ajustements ansible en environnement docker (reboot ...)

```
when: ansible_virtualization_type is defined and ansible_virtualization_type=="docker"
```

Ressources

- Docker docs.docker.com/engine/installation/
 - image docker-ubuntu14-ansible-docker hub.docker.com/r/arnaudblancher/docker-ubuntu14-ansible-docker/
- Jmeter 2.13 archive.apache.org/dist/jmeter/binaries/apache-jmeter-2.13.zip
- Gitlab about.gitlab.com/downloads/
- Ansible Galaxy galaxy.ansible.com/
 - rôle docker_provisionning galaxy.ansible.com/arnaudblancher/docker_provisionning/
 - rôle jmeter galaxy.ansible.com/arnaudblancher/jmeter/
 - rôle docker_unprovisionning galaxy.ansible.com/arnaudblancher/docker_unprovisionning/

Code source de la démonstration :