

Distribution EOLE - Tâche #35384

Scénario # 35185 (Terminé (Sprint)): EOLE 2.9 : L'utilisation des tag pour les images de conteneur n'est pas suffisante pour s'assurer de la mise à jour

Tester

10/05/2023 11:42 - Benjamin Bohard

Statut:	Fermé	Début:	01/10/2022
Priorité:	Normal	Echéance:	
Assigné à:	Ludwig Seys	% réalisé:	100%
Version cible:	Livraison Cadoles 31/05/2023	Temps estimé:	0.00 heure
		Temps passé:	0.00 heure
Description			
Les changements affectent Query-Auto et Maj-Auto qui doivent exécuter le script ajouté dans /usr/share/eole/majauto_pre.			
Pour avoir des images à mettre à jour, le plus simple est sans doute d'en installer des supplémentaires depuis un registre local.			

Historique

#1 - 10/05/2023 11:47 - Benjamin Bohard

- Fichier Dockerfile ajouté
- Fichier build_and_register.sh ajouté
- Fichier main.yml ajouté

#2 - 10/05/2023 11:51 - Benjamin Bohard

Pour utiliser un registre autre que le hub eole, il faut patcher le template podman_registries.conf par exemple, en ajoutant à la fin :

```
[[registry]]
location=<l'url du registre>
insecure=true
```

#3 - 10/05/2023 14:12 - Ludwig Seys

- Assigné à mis à Ludwig Seys

#4 - 11/05/2023 11:32 - Ludwig Seys

- Statut changé de Nouveau à En cours

#5 - 11/05/2023 13:44 - Ludwig Seys

- Statut changé de En cours à Résolu
- % réalisé changé de 0 à 100

TEST VALIDE, les fonctionnalités attendues sont présentent.

méthodologie.

montage du registry dans une debian instancié dans le réseau académie.

```
apt install podman
podman pull registry:2
```

```
podman run -d -p 8002:5000 docker.io/library/registry:2
podman ps #(pour valider le lancement)
```

montage d'une eolebase pour build des images.
copier le script et le Dockerfile.

```
vi build_and_register.sh
registry="192.168.0.103:8002" => indiquer l'adresse et le port d'écoute sur le debian
```

&&

```
vi /etc/containers/registries.conf
```

ajouter en fin :

```
[[registry]]
location= "192.168.0.103:8002"
insecure= true
```

placer le Dockerfile et le script.
pour régénérer une image changer la valeur de : env TZ=
puis lancer le script pour générer l'image.

note : il est intéressant de modifier le dockerfile pour limiter le nombres de messages.

=>

```
from ubuntu:jammy
env TZ='Brazil/Acre'
cmd while true; do sleep 10; echo $(date); done
```

dans un amon a jour :

```
cp /usr/share/eole/creole/{distrib,modif}/podman_registries.conf
vi /usr/share/eole/creole/modif/podman_registries.conf
```

ajouter en fin :

```
[[registry]]
location = "192.168.0.103:8002" #l'adresse du registry
insecure = true
```

faire ensuite :

```
gen_patch
CreoleCat -t podman_registries.conf
```

tirer l'image (valide la conf)

```
podman pull 192.168.0.103:8002/test:dev
```

avec un podman ps récupérer l'id et le lancer:

```
podman run -d fd6149e36b67e7042451aae4b7520605654a80267d11180e9e1c0bce1c6becd4
```

un journalctl -f doit afficher l'heure dans le fuseau du conteneur.

--- ensuite générer une seconde image en changeant la donnée env TZ='Brazil/Acre'.
régénérer l'image et la pousser sur le registry (le script suffit)
et lancer un query-auto depuis l'amon

sans nouvelles images généré le message suivant s'annonce :

Gestion des images de conteneur

Toutes les images sont à jour

avec de nouvelles images il indique l'image en question.

un maj-auto télécharge la nouvelle image.

faire un podman rm -f id-conteneur
et lancer un run de la nouvelle.

un journalctl permet d'afficher le nouveaux fuseau horaires.

créer un service, pour arrêter et démarrer le conteneur, en cas de nouvelles images et d'un arrêt/redémarrage du service il lance la dernière image:
vi /etc/systemd/system/test.service

```
# container-b90133ce6e521d935ff6e55365441445eaf563d67001cf4712a37523222018fe.service  
# autogenerated by Podman 3.4.4  
# Thu May 11 11:18:41 CEST 2023
```

```
[Unit]  
Description=Podman container-b90133ce6e521d935ff6e55365441445eaf563d67001cf4712a37523222018fe.service  
Documentation=man:podman-generate-systemd(1)  
Wants=network-online.target  
After=network-online.target  
RequiresMountsFor=/run/containers/storage
```

```
[Service]  
Environment=PODMAN_SYSTEMD_UNIT=%n  
Restart=on-failure  
TimeoutStopSec=70  
ExecStartPre=/bin/rm -f %t/%n.ctr-id  
ExecStart=/usr/bin/podman run --cidfile=%t/%n.ctr-id --sdnotify=common -d 192.168.0.103:8002/test:dev  
ExecStop=/usr/bin/podman stop -t 10 --cidfile=%t/%n.ctr-id  
ExecStopPost=/usr/bin/podman stop -t 10 --cidfile=%t/%n.ctr-id  
Type=forking
```

```
[Install]  
WantedBy=default.target
```

faire un daemon-reload pour activer le service.

#6 - 22/05/2023 09:49 - Joël Cuissinat

- Statut changé de Résolu à Fermé

- Restant à faire (heures) mis à 0.0

Non re-testé ;)

Fichiers

Dockerfile	81 octets	10/05/2023	Benjamin Bohard
build_and_register.sh	300 octets	10/05/2023	Benjamin Bohard
main.yml	341 octets	10/05/2023	Benjamin Bohard