VMoodle : Guide technique

Principe de virtualisation

La virtualisation de Moodle consiste à n'utiliser qu'une seule base de code installée pour opérer un nombre indéterminé de plates-formes Moodle, autonomes ou configurées en réseau. Le procédé utilise une mise en base de données (dynamique) des paramètres de configuration principaux (chemins et base de données) des instances virtuelles, et de procéder à une commutation très précoce de la configuration de service à partir de ce registre.

La commutation de configuration est basée sur la reconnaissance du nom d'hôte dans les variables d'environnement du serveur. Le nom d'hôte présenté par le serveur Web à l'environnement d'exécution PHP doit correspondre à l'identité d'hôte (wwwroot virtuel) stocké dans le registre. De ce fait, les cas d'usage suivant nécessitent des précautions et des mises en oeuvre particulières :

- Utilisation de scripts CLI en mode console (voir ci-dessous)
- Utilisation de plusieurs Moodle sur la même racine de domaine (en sous-répertoires)

Contraintes sur les scripts CLI

Les scripts CLI standard de Moodle ne sont pas compatibles avec la virtualisation. En effet, en l'absence d'environnement serveur définissant le nom de domaine activé, les scripts CLI standard sont incapables de procéder à la résolution de virtualisation.

C'est pourquoi l'implémentation VMoodle vous fournit la plupart des scripts CLI standard adaptés à un usage virtualisé par l'ajout d'un paramètre de ligne de commande –host permettant d'expliciter cette résolution :

```
php {cliscript} --{param1}={val1} --host=http://virtuel.monmoodle.fr
```

VMoodle fournit également une série de scripts d'industrialisation pour procéder massivement à des déploiements et des transformations de nombreuses unités Moodle.

Scripts standards disponibles

- local/vmoodle/cli/automated_backups.php ⇒ admin/cli/automated_backups.php
- local/vmoodle/cli/maketestplan.php ⇒ admin/tool/generator/cli/maketestplan.php
- local/vmoodle/cli/mysql_collation.php ⇒ admin/cli/mysql_collation.php
- local/vmoodle/cli/mysql_compressed_rows.php ⇒ admin/cli/mysql_compressed_rows.php
- local/vmoodle/cli/mysql_compressed_rows.php ⇒ admin/cli/mysql_compressed_rows.php
- local/vmoodle/cli/mysql_engine.php ⇒ admin/cli/mysql_engine.php
- local/vmoodle/cli/purge_caches.php ⇒ admin/cli/purge_caches.php
- local/vmoodle/cli/replace.php ⇒ admin/cli/replace.php
- local/vmoodle/cli/reset_admin.php ⇒ admin/cli/reset_admin.php
- local/vmoodle/cli/reset_password.php ⇒ admin/cli/reset_password.php
- local/vmoodle/cli/schedule_task.php ⇒ admin/cli/schedule_task.php
- local/vmoodle/cli/update_langpacks.php ⇒ admin/cli/update_langpacks.php

- local/vmoodle/cli/upgrade.php ⇒ admin/cli/upgrade.php
- local/vmoodle/cli/upgrade_assignments.php ⇒ admin/cli/upgrade_assignments.php

Scripts d'industrialisation

- local/vmoodle/cli/bulkcreatenodes.php
- local/vmoodle/cli/bulkdestroynodes.php
- local/vmoodle/cli/bulkmysqlcollation.php
- local/vmoodle/cli/bulkmysqlcompressedrows.php
- local/vmoodle/cli/bulkmysqlengine.php
- local/vmoodle/cli/bulkpurgecaches.php
- local/vmoodle/cli/bulksnapshot.php
- local/vmoodle/cli/bulkupdatelangpacks.php
- local/vmoodle/cli/bulkupgrade.php

Scripts additionnels

Sur la base de cette structure de lancement industrialisé, d'autres scripts spécifiques VMoodle sont disponibles dans le répertoire `/local/vmoodle/cli`

Lanceur de commande de superadministration

VMoodle dispose d'un jeu de commandes de super-administration réseau pour lancer des actions sur tout ou partie des instances moodle virtualisées. Ces commandes peuvent être invoquées à partir de l'interface d'administration VMoodle, mais aussi désormais en ligne de commande par la commande :

```
php local/vmoodle/cli/run_command.php
```

La commande s'exécutera toujours en partant de l'instance principale (on ne peut lancer une commande de super-administration dans une instance virtuelle). La commande accepte 4 paramètres :

- **fromhost** : si la commande doit effectuer une copie de données à partir d'une instance de référence (synchro de config, copie de table, copie de fichiers, etc), alors on donnera l'identité (wwwroot) de cette instance. En l'absence de paramètre c'est la plate-forme principale qui sera prise comme 'source'.
- **tohosts** : une liste à virgule des identités d'instances destinataires de la commande. Les cibles sont identifiées par leur wwwroot. Alternativement, `tohostsmatch` peut être utilisée avec un motif REGEXP pour sélectionner un ensembles d'instances par filtrage de nom.
- **command** : Le nom de la classe-commande (voir Liste des classes-commandes en annexe)
- **attributes** : une liste de clefs/valeurs formatée comme une QUERYSTRING http.

Exemple de commande :

```
 \$sudo -u www-data /usr/bin/php local/vmoodle/cli/run_command.php
     --fromhost=http://source.virtual.moodle.org
     --
tohosts=http://target1.other.moodle.org,http://target2.other.moodle.org
```

```
    --command=generic/CopyFilearea --
attributes=fileareaid=mod_h5p%2Flibraries%3A0
```

Test des hôtes cibles sur une cible "tohostsmatch"

Utilisation de plusieurs Moodle sur la même racine d'hôtes (sous-répertoires)

L'utilisation de la clef supplémentaire de configuration :

```
 $CFG->vmoodleusesubpath = true;
```

Permet d'activer la configuration 'sous-répertoires' de VMoodle. Toutes les instances virtuelles seront alors opérées comme un premier niveau de sous-répertoire d'une raçine de domaine unique :

```
https://mon.vmoodle.org/moodle1
https://mon.vmoodle.org/moodle2
https://mon.vmoodle.org/moodle3
...
```

L'avantage de ce modèle est l'économie en termes de certificats HTTPS (un seul certificat de domaine pour toutes les instances). Il peut par contre provoquer une certaine confusion dans la mémorisation de données saisies côté client web.

Le nom d'instance est automatiquement capté à partir du nom du sous-répertoire.

La mise en place de moodle virtualisés à un niveau inférieur de l'arborescence n'est pas supportée.

Pour mettre en place ce type il sera nécessaire d'assurer la mise en place de liens symboliques d'instance dans le répertoire d'installation du code de moodle :

```
 drwxrwsr-x 11 root root  admin
 drwxrwsr-x 24 root root  auth
 drwxrwsr-x  6 root root  availability
 drwxrwsr-x  7 root root  backup
 drwxrwsr-x  6 root root  badges
 -rw-rwSr--  1 root root  behat.yml.dist
 drwxrwsr-x 70 root root  blocks
 drwxrwsr-x  3 root root  blog
 -rw-rwSr--  1 root root  brokenfile.php
 drwxrwsr-x  6 root root  cache
 drwxrwsr-x  6 root root  calendar
 drwxrwsr-x  4 root root  cohort
 drwxrwsr-x  4 root root  comment
 drwxrwsr-x  4 root root  competency
 drwxrwsr-x  5 root root  completion
 ...
 lrwxrwxrwx  1 root root  moodle1 => .
 lrwxrwxrwx  1 root root  moodle2 => .
 lrwxrwxrwx  1 root root  moodle3 => .
```

```
...
```

Cette mise en place assure à chaque moodle virtualisé en sous-répertoire la disponibilité du code source de moodle. Les outillages de gestion des instances gèrent la mise en place de ces liens symboliques pour le compte de l'administrateur Moodle à travers une procédure locale de sudo. La configuration VMoodle doit désigner un utilisateur sudo ayant la possibilité de manipuler le répertoire d'installation de moodle, à partir de l'utilisateur titulaire de l'exécution du serveur (apache, httpd, nginx, ou autre suivant l'environnement).

Revenir à l'index du composant - Revenir à l'index des plugins - Revenir au catalogue

From:
https://docs.activeprolearn.com/ - **Documentation Moodle ActiveProLearn**

Permanent link:
https://docs.activeprolearn.com/doku.php?id=local:vmoodle:technique&rev=1567068881

Last update: **2025/10/15 10:35**