Formation Zéphir - Personnalisation d'un module

Académie de Rennes

Emmanuel Garette

Cadoles

18 mars 2016

Cadoles

Plan

Cadoles

EOLE

- EOLE : Ensemble Ouvert Libre et Evolutif ;
- projet national de serveurs pour les établissements scolaires et académiques ;
- depuis 2000 ;
- basé sur Ubuntu depuis 2007 sous forme de modules ;
- objectifs :
  - utilisation de logiciels libres ;
  - modulaire (évolutif, ouvert adaptable) ;
  - facile à mettre en œuvre et à déployer ;
  - administrable à distance ;

Nouveautés 2.4

- Basé sur la distribution Precise Pangolin (12.04) ;
- support jusqu'en 2017 ;
- réorganisation reconfigure/instance ;
- nouveau cœur : tiramisu ;
- plus d'informations dans les dictionnaires ;
- firewall :
  - géré dans les dictionnaires ;
  - filtrage des flux entrants uniquement ;
- gen_config :
  - interface web ;
  - identique sur Zéphir ;
  - plus de "validation" par onglet ;
- nouvelles commandes :
  - CreoleGet et CreoleSet ;
  - CreoleService et CreoleRun ;
  - CreoleLock.

Cadoles

Nouveautés 2.4

- Plus de mise à jour minimum/complète mais différentes releases sont proposées ;
- les paquets sont disponibles à l'adresse http ://eole.ac-dijon.fr ;
- nouveau format du fichier /etc/eole/config.eol ;
- nouvelle sous arborescence : /usr/share/eole/sbin.

Nouveautés 2.5

- Basé sur la distribution Trusty Tahr (14.04) ;
- support jusqu'en 2019 ;
- peu de changement dans le coeur ;
- possibilité de partitionnement manuel ;
- à partir de 2.5.2 : mot de passe généré à la 1er instance ;
- quelques changement de logiciel :
  - Bacula => Bareos,
  - Dansguardian => e2guardian.

Plan

Cadoles

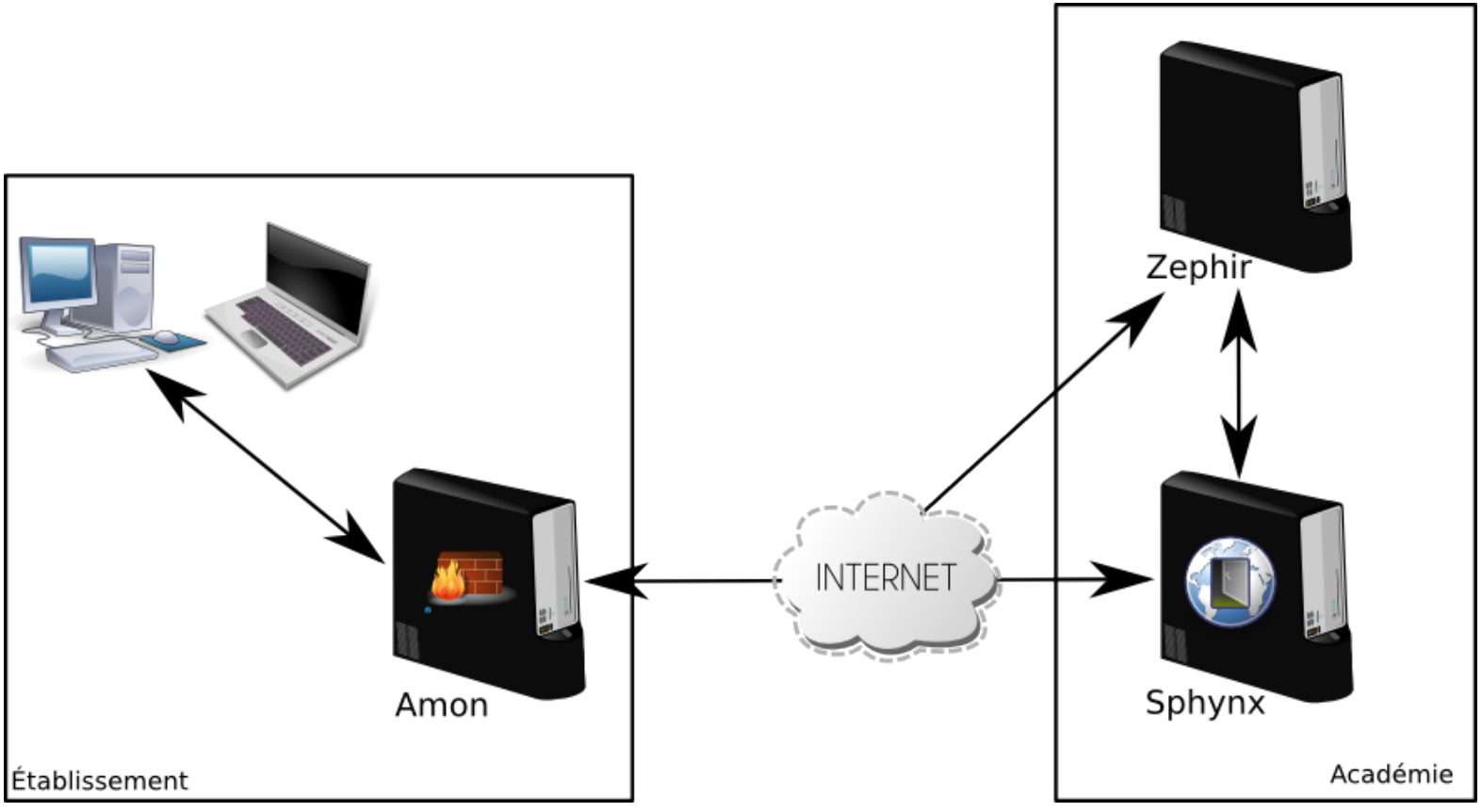
Amon

- Amon, la passerelle pare-feu :
  - partage des sous-réseaux et connexion à internet (pare-feu) ;
  - authentifications des utilisateurs ;
  - réseau virtuel privé ;
  - cache web ;
  - reverse proxy web.

Sphynx

- Sphynx, concentrateur pour réseau privé virtuel :
  - relier des réseaux privés entre eux (RVP) ;
  - possibilité de haute disponibilité.

Schéma



Cadoles

Scribe

- Scribe, le serveur de fichiers pédagogique :
  - partage de fichiers (avec quotas, ACL) ;
  - partage d'imprimantes ;
  - gestion de comptes utilisateurs et de groupes ;
  - gestion des accès utilisateurs ;
  - gestion des postes clients ;
  - gestion des élèves (devoirs, observations, ...) ;
  - serveur web avec portail web (Envole).

Horus

- Horus, le serveur de fichiers administratif :
  - partage de fichiers (avec quotas, ACL) ;
  - partage d'imprimantes ;
  - gestion de comptes utilisateurs et de groupes ;
  - gestion des accès utilisateurs ;
  - gestion des postes clients ;
  - applications nationales.

AmonEcole

- AmonEcole, solution de conteneurs cumulant les fonctionnalités de :
  - Amon,
  - Scribe.

Zéphir

- Zéphir, gestion du parc des serveurs EOLE :
  - déploiement et gestion d'un parc de serveurs,
  - gestion de la configuration des serveurs,
  - surveillance et lancement d'actions à distance.

Seshat

- le relais de messagerie pour les domaines intra-académiques des Scribe ;
- annuaire centralisé avec alimentation décentralisée.

Thot

- annuaire centralisé avec alimentation centralisée.

Hapy

- serveur de solution virtuelle un nœud ou multiple nœuds.

Plan

La phase d'installation

- Pour installer un module, il suffit de :
  1. démarrer sur l'iso téléchargée sur le site d'EOLE (gravée sur CD-ROM ou copiée sur clé USB),
  2. sélectionner le module à installer parmi ceux proposés et valider,
  3. possibilité de partitionner le serveur (obligatoire sur eolebase et avec serveur ayant plus d'un disque),
  4. valider le bouton *continuer* à la fin de l'installation,
  5. ouvrir une session avec l'utilisateur *root* et le mot de passe par défaut présenté à l'écran,
  6. en mode conteneurs, générer les conteneurs avec la commande *gen_conteneurs*.

La phase de configuration

- Il faut une bonne connaissance du réseau local ;
- en mode autonome :
  - lancer la commande `gen_config`,
  - configurer le serveur,
  - sauvegarder le fichier ;
- en mode Zéphir :
  - configuration dans l'application Zéphir : https ://<nom_du_serveur> :8070/ ou via l'application de configuration du module (`gen_config`) ;
  - enregistrement au Zéphir ;
  - descente ou montée de la configuration.

La phase d'instanciation

- Lancer la commande `instance` ;
- renseigner les mots de passe ;
- mise à jour ;
- éventuellement redémarrage.

Les mots de passe

- Au premier lancement de l'instanciation, il est nécessaire de modifier les mots de passe :
  - *root*,
  - administrateur à droits restreints *eole*,
  - possibilité d'ajouter plusieurs administrateurs à droits restreints,
  - sur Amon, en cas d'utilisation d'un réseau pédagogique et administratif, un second administrateur (*eole2*) permet d'administrer le réseau pédagogique,
  - *admin* sur Scribe et Horus,
  - *admin_zephir* sur Zéphir.

Les mots de passe

Par défaut, le système vérifie la complexité des mots de passe. Pour cela, il utilise un système de *classes de caractères* en combinaison avec un nombre de caractères minimum :

- il faut combiner des caractères issus de différentes classes :
  - les lettres en minuscule,
  - les lettres en majuscule,
  - les chiffres,
  - les caractères spéciaux (Ex : $*ù%£,;:!§/.?).
- il faut un minimum de caractères selon le nombre de classes utilisées :
  - une seule classe de caractères : impossible,
  - deux classes de caractères : 9 caractères minimum,
  - trois et quatre classes : 8 caractères minimum.

Cette configuration est, bien évidemment, modifiable dans l'interface de configuration du module, gen_config, en mode expert.

La phase d'administration

- Correspond à l'exploitation du serveur ;
- chaque module possède des fonctionnalités propres, souvent complémentaires ;
- diverses interfaces permettent la mise en œuvre de ces fonctionnalités et en facilitent l'usage :
  - l'interface d'administration EOLE (EAD, EOP, ...),
  - l'interface semi-graphique,
  - diverses interfaces d'administration (Zéphir-web, CUPS, Sympa, ...),
  - différents outils (Era, ...) ;
- gestion des mises à jour.

Utilisation de VirtualBox

- Ajouter l'image Scribe ;
- choisir "Réseau privé hôte" dans le "Mode d'accès réseau" dans Préférence/Réseau/Carte 1 ;
- faire un instantané.

Configuration autonome

- Présentation de l'interface :
  - accessible après instanciation sur le port 7000 (si autorisé),
  - nécessite une authentification,
  - barre de menu,
  - les familles,
  - la partie centrale,
- les modes :
  - basique,
  - normal,
  - expert,
  - assitant,
  - debug.

Configuration autonome

- Ouverture/enregistrement :
  - ne peut plus choisir le nom du fichier "config.eol",
  - possibilité d'importer/exporter un fichier (astuce permet de déverouiller toutes les variables),
  - page des variables obligatoires non renseignées,
  - visualisation des différences avant l'enregistrement,
  - enregistrement dans le fichier "config.eol" ;
- les variables :
  - Les variables typées (caractère, chiffre, ...),
  - les variables par défaut/modifié,
  - les utilisateurs,
  - les variables obligatoires,
  - les variables calculées,
  - les variables verrouillées automatiquement,
  - les variables sauvegardées automatiquement,
  - les variables multi,
  - les variables groupées,
  - les variables avec choix (imposés ou ouverts)

Pratique

- Configurer le serveur via la commande gen_config ;
- informations utiles :
  - Adresse ip de la carte eth0,
  - Passerelle,
  - DNS : DNS de la machine hôte ;
- désactiver l'anti-virus ;
- instancier le serveur via la commande instance.

Plan

Instance ou reconfigure

- L'instance ne doit être lancée qu'une seule fois sur les modules ;
- en cas de mise à jour, d'installation de paquet ou de changement de paramétrage : `reconfigure`.

Version/release et dépôt

- La version : correspond à la version du socle de base (version d'Ubuntu) ;
- la release :
  - reçoit les mises à jour de sécurités,
  - reçoit les corrections des dysfonctionnements critiques,
  - mais chaque release à un niveau fonctionnel différent ;
- les erratas : https ://dev-eole.ac-dijon.fr/projects/modules-eole/wiki/Errata ;
- les dépôts :
  - les releases stables,
  - les releases candidates,
  - la version de développement (future release).

Les mises à jour

- Procédure :
  - par l'EAD,
  - par l'interface semi-graphique,
  - par Zéphir,
  - à la ligne de commande :
    - `Query-Auto` : liste les mises à jour,
    - `Maj-Auto` : lance la mise à jour sans question,
    - `Query-Cd` : liste les mises à jour sur un CD,
    - `Maj-Cd` : mise à jour grâce à un CD,
    - `Maj-Release` : mise à jour vers une release supérieure,
    - `Upgrade-Auto` : mise à jour vers une version supérieure,
  - automatiquement via une tâche hebdomadaire "post" sauvegarde de eole-schedule.

Mise à jour via un CD/clef USB

Le but est de pouvoir mettre à jour un serveur sans utiliser la bande passante de l'établissement.

- Création du cache dans l'académie : `Maj-Auto --download` ;
- copie de /var/cache/apt/archives/*.deb sur un support amovible ;
- sur le serveur en établissement, copier le contenu du support dans /var/cache/apt/archives/ ;
- lancer `Maj-Auto`.

Diagnostic

- La commande `diagnose` permet de tester différents points du serveur ;
- La commande `diagnose -L` propose un diagnostic étendu.

Pratique

- Faire un `reconfigure` ;
- lancer un `diagnostic` ;
- lancer un `diagnostic` étendu.

Firewall

- Sur tous les modules ;
- géré soit par Creole soit par Era ;
- activé par bastion ;
- configure firewall et tcpwrapper ;
- si Era n'est pas installé, possibilité de désactiver ;
- pour ouvrir le firewall ouvre.firewall ;
- recharger les règles de firewall : service bastion restart ;
- pour connaître la liste des règles : iptables-save.

Schedule

- Gestion des tâches planifiées avec ou sans sauvegarde ;
- tâche "pre" sauvegarde ;
- tâche "post" sauvegarde ;
- tirage au sort à l'instance de :
  - l'heure de démarrage entre 1h et 6h,
  - le jour dans la semaine d'exécution des tâches hebdomadaires et mensuelles (le 1er xxx du mois) ;
- pour lister : /usr/share/eole/sbin/manage_schedule -l .

Pratique

- Vérifier les règles ;
- ouvrir le firewall et revérifier les règles ;
- refermer le firewall ;
- lister les tâches du schedule, désactiver une tâche et la réactiver ;
- activer la sauvegarde pour cette nuit.

Onduleur

- Gestion des onduleurs ;
- utilisation du logiciel libre NUT ;
- maître - esclave.

Pratique

- Configurer de l'onduleur dans gen_config :
  - onglet "services", activer NUT,
  - onglet "onduleur ajouter un nouvel onduleur maitre :
    - Nom de l'onduleur : dummy
    - Port de l'onduleur : dummy.dev
    - Pilote : dummy-ups
- enregistrer la configuration
- créer le fichier : /etc/nut/dummy.dev avec :
  - battery.charge : 50
  - ups.status : OL
  - TIMER 10
  - ups.status : OB
- reconfigurer, toutes les 10 secondes le status change d'état.

Commande distante via SSH

- Protocole de communication sécurisée ;
- possibilité d'interdire la connexion ssh à root ;
- possibilité d'interdire la connexion sans clef rsa ;
- choix des adresses autorisées.

Utiliser ssh depuis GNU/Linux

- Ouvrir une connexion : `ssh utilisateur@ip_serveur` ;
- ouvrir une connexion pour commande graphique : `ssh -X utilisateur@ip_serveur` ;
- transfert de fichier : `scp fichier.txt utilisateur@ip_serveur:.`

Utiliser ssh depuis Windows

- Installer putty, xming et winscp ;
- configuration du putty :
  - Terminal/Feature/Disable application keypad mode => coché,
  - Window/Translation/Received data assumed to be in which character set => UTF-8,
  - Connection/SSH/X11/Enable X11 forwarding => coché,
  - sauvegarder comme "Default Settings" ;
- vérifier la présence de xming.

Screen

- Si la connexion se coupe, impossible de récupérer la session ;
- screen permet de détacher un shell et de le récupérer ;
- pour le lancer : `screen` ;
- pour détacher : *ctrl a d* ;
- attention déconnexion automatique : `unset TMOUT` ;
- pour reprendre un screen : `screen -rd`.

Pratique : connexion ssh

- Activer la connexion juste pour votre IP ;
- se connecter sur le serveur en ssh ;
- copier un fichier ;
- lancer `gen_config` à travers ssh ;
- utiliser `screen`.

Trouver de l'information sur le serveur

- Les journaux sont remontés sur le maître en mode conteneur ;
- les journaux locaux sont maintenant dans /var/log/rsyslog/local/ ;
- principaux journaux :
  - kernel/ : tous les messages du noyau ;
  - auth/ : contient les connexions des utilisateurs ;
  - cron/ : fichier log du service cron (planificateur système).
- diagnose ;
- gen_rpt.

Trouver des informations sur Internet

- la documentation ;
- les FAQ ;
- les archives des listes de diffusion ;
- le Wiki EOLE ;
- recherche sur Internet ;
- équipes d'assistance académiques.

Plan

Gestion des utilisateurs

- Utilisation d'un serveur LDAP local ou distant ;
- LDAP local : scripts (/usr/share/zephir/utils) :
  - add_user.py,
  - del_user.py,
  - list_users.py ;
- un utilisateur = des affectations.

Modules, établissements et serveurs

- Module : distribution spécifique + variantes ;
- établissement scolaire : lieu physique ;
- serveur : un module dans un établissement.

Enregistrement Zéphir

- Processus obligatoire et manuel ;
- création dans la base + lien sécurisé ;
- lancer : enregistrement_zephir :
  - connexion au réseau,
  - connexion à Zéphir,
  - création ou choix du serveur,
  - gestion des configurations :
    - nouveau serveur : "récupérer les fichiers de la variante",
    - serveur configuré en mode autonome : "sauver la configuration actuelle",
    - serveur configuré en mode Zéphir : "utiliser la configuration définie sur Zéphir".

Pratique : configuration Zéphir du serveur

- Création d'un nouvel utilisateur ;
- affecter tous les droits à l'utilisateur ;
- se reconnecter avec ce nouvel utilisateur ;
- ajouter un nouvel établissement ;
- enregistrement du serveur sur le Zéphir en remontant la configuration.

Sauvegarde Zéphir

- sauvegarde de Zéphir : sauvegarde.sh ;
- restauration du Zéphir : restauration.sh ;
- emplacement des sauvegardes : /var/lib/zephir_backups/ (avec la date).

Pratique

- Lancer une sauvegarde du Zéphir ;
- restaurer le serveur avec la sauvegarde réalisée.

Préférence des utilisateurs

- Contient les informations sur l'utilisateur (nom, prénom et mail) ;
- activer les alertes par mail ;
- envoyer une clef SSH.

Pratique

- Créer et configurer l'utilisateur "formation" ;
- créer une clef SSH :
  1. *ssh-keygen -t rsa*,
  2. récupérer le fichier /.ssh/id_rsa.pub.
- ajouter la clef à l'utilisateur.

Groupe de serveurs

- Les variantes ne servent qu'à la configuration initiale ;
- changement sur plusieurs serveurs : groupe de serveurs ;
- sélectionner les serveurs suivant différents critères ;
- sauvegarde du groupe ;
- possibilité d'ajout ultérieur de machine ;
- actions supplémentaires.

Pratique

- Créer un groupe de serveur ;
- autoriser l'utilisateur à se connecter par clef SSH.

Surveillance des serveurs

- Sur la page d'accueil ;
- sur chaque serveur : page Etat ;
- surveillance des serveurs ;
- état d'un groupe de serveurs.

Alertes

- Envoi d'un mail en cas de détection d'un problème :
  - remontée par le serveur d'une erreur,
  - un serveur n'a pas contacté Zéphir,
  - une opération s'est mal déroulée ;
- l'utilisateur doit avoir configuré son adresse mail ;
- à la sélection d'un groupe enregistré : cocher "surveiller ce groupe".

Pratique

Surveiller un groupe.

Actions

- Zéphir agit à distance par des actions ;
- elles sont mises en file d'attente ;
- c'est le serveur qui se connecte au Zéphir ;
- exemple d'action : envoi de la configuration, mise à jour, sauvegarde de l'état actuel du serveur.

Pratique

- Lancer l'action de remontée de la configuration ;
- forcer la synchronisation : synchro_zephir ;
- restaurer le serveur dans VirtualBox et redescendre la configuration.

Plan

Lire un fichier

- Lire un fichier complet : *less*, *cat* ;
- lire le début d'un fichier : *head* ;
- lire la fin d'un fichier : *tail* (avec option -f) ;

Editer un fichier : *vim*

- Les modes :
  - le mode normal : *Esc* ;
  - le mode insertion : *i* ;
  - le mode visuel : *v*.
- Les fichiers :
  - *:e fichier.txt* : éditer le fichier ;
  - *:w* : écrire dans le fichier ;
  - *:q* : quitter le fichier.
- Autre :
  - annuler, refaire : *u* ; *ctrl r* ;
  - couper, copier, coller : *d*, *y*, *p* ;
  - caractère, mot, ligne : *l*, *w*, *1*.

Pratique

- lire la fin du fichier /var/log/rsyslog/local/auth/auth.info.log ;
- lire le début du fichier /var/log/rsyslog/local/rsyslog/rsyslog.info.log ;
- éditer le début du fichier /var/log/rsyslog/local/rsyslog/rsyslog.info.log :
  - passage d'un mode à un autre ;
  - taper un peu de texte ;
  - faire des couper/copier/coller ;
  - annuler/refaire.

Création de patch

- Génération normal : /usr/share/eole/creole/distrib => /etc
- Génération avec patch : /usr/share/eole/creole/distrib + /usr/share/eole/creole/patch => /var/lib/creole => /etc
- Créer un patch :
  - copie du fichier original /usr/share/eole/creole/distrib => /usr/share/eole/creole/modif ;
  - modification du fichier dans /usr/share/eole/creole/modif ;
  - exécution "gen_patch" ;
  - reconfiguration.
- Désactiver un patch : supprimer le fichier dans /usr/share/eole/creole/patch.

Pratique

- Modifié dans le fichier de configuration de proftpd.conf : UseReverseDNS de "off" à "on" ;
- créé le patch.

Dictionnaire Creole

- Un dictionnaire est un fichier XML Creole ;
- décrit les variables présente dans l'interface de configuration ;
- informations utiles pour les services ;
- les dictionnaires locaux sont dans /usr/share/eole/creole/dicos/local/.

CreoleLint

- Vérifie la syntaxe du dictionnaire : `CreoleLint -d` ;
- test la génération d'un template suivant contrainte : `CreoleLint -t nom_du_template`

CreoleCat

- Templatise un fichier sans faire de reconfigure (objectif de test) ;
- `CreoleCat -t nom_du_template`
- `CreoleCat -t nom_du_template -o /tmp/template.tmpl`
- `CreoleCat -s /tmp/nom_du_template -o /tmp/template.tmpl`

Dictionnaire Creole : les paquets

- Les paquets fonctionnent sur le maitre ou dans un conteneur ;
- permet d'installer les paquets sans passer par les dépendances de paquet.

Dictionnaire Creole : les services

- Services fonctionnent sur le maitre ou dans un conteneur ;
- différentes méthodes : service, upstart ou apache.

Pratique

- Installer le paquet ircd-irc2 ;
- ajouter un dictionnaire pour gérer :
  - l'installation du paquet,
  - le service.

Dictionnaire Creole : le firewall

- On commence par ouvrir les accès (service_access) ;
- on ajoute ensuite les restrictions (service_restiction).

Pratique

- Ajouter des règles de firewall et tcpwrapper pour autoriser les accès.

Dictionnaire Creole : les familles

- Les variables sont classés dans des familles.

Dictionnaire Creole : les variables

- Les variables sont typées :
  - number, port,
  - string, mail, filename, unix_user, web_address,
  - ip, local_ip, netmask, network, broadcast,
  - nebios, domain, domain_strict, hostname, hostname_strict,
  - oui/non, on/off, yes/no.

Pratique

- Dans la famille "services" ajouter la variable "activer_ircd" ayant pour valeur possible "oui" et "non" ;
- tester gen_config.

Dictionnaire Creole : les fichiers templates

- Le nom des fichiers templates sont dans les dictionnaires (file) ;
- le répertoire de destination doit existé par défaut ;
- possibilité de définir les droits sur le fichier ;
- nom de la source peut être différents de la destination.

Pratique

- Ajouter le template /etc/ircd/ircd.conf.

Dictionnaire Creole : les contraintes

- pour tester/remplir/grouper/conditionner des variables ;
  - calcul automatique avec fonction personnalisé (fill),
  - calcul automatique non modifiable (auto),
  - choix de réponse (valid_enum),
  - cacher des variables suivant des contraintes (condition),
  - groupe de variables (group) :

Pratique

- Ajouter une famille "ircd" ;
- configuration de la M line :
  M :<ircd_domaine> ::<ircd_libelle> :<ircd_port> :000A :
  - le nom de domaine est une copie de web_url,
  - le libellé est la concaténation de "Serveur IRC pour" et libelle_etab et n'est pas modifiable par l'utilisateur,
  - le port ircd est par défaut le port 6667 modifiable en mode expert,
  - modifier la balise service_access ;
- pouvoir ajouter des K line :
  K :<ircd_banned_hostname :ircd_banned_comment :* :0 :
  - variable groupé avec possibilité de ne pas mettre de valeur,
  - nom d'hôte ou domaine banni,
  - proposer un commentaire par défaut.

Pratique

- Faire une condition pour désactiver la famille et le service ;
- tester gen_config ;
- mettre la valeur de ircd_domaine à localhost.localdomaine.

Dictionnaire Creole : l'aide

- Afficher de l'aide dans l'interface : <help></help>.

Pratique

- Ajouter de l'aide sur la variable activer_ircd.

Langague de template Creole

- Un template utilise le langage de template Creole.
- variable Creole préfixé par

  %%
- Test :

  ```
  %if EXPRESSION code if %else code else %end if
  ```
- Boucle :

  ```
  %for %%i in EXPRESSION hello %%i %end for
  ```
- Boucle pour variable multi évolué avec esclave :

  ```
  %for %%var in %%variablemaitre

  %%var, %%var.esclave1, %%var.esclave2

  %end for
  ```

Langague de template Creole

- Test existance :

  ```
  %if %%is_defined('variable') code if %end if
  ```

- valeur vide :

  ```
  %if %%is_empty(%%variable) code if %end if
  ```

- nom de variable dans une variable :

  ```
  %set %%var='adresse_ip_eth0' %%getVar(%%var)
  ```

Pratique 1

- Configuration de la "M line :
  M :<ircd_domaine> ::<ircd_libelle> :<ircd_port> :000A" ;
- configurer les "K line :
  K :<ircd_banned_hostname :ircd_banned_comment :* :0 :".

Pratique 2

- Templatisé le fichier /etc/ircd/ircd.motd ;
- test l'existance de la variable web_url et fait un message personnalisé si existe ;
- créer une variable ircd_motd_message avec trois valeurs : "message 1", "message 2" et "personnalisé" ;
- faire deux messages différents pour "message 1" et "message 2" ;
- créer une variable ircd_motd_personnalise visible si ircd_motd_message est "personnalisé" ;
- valider que ircd_libelle soit différent de ircd_motd_message.

Les fonctions personnalisées

- Ajouter une fonction utiliser dans un dictionnaire : /usr/share/creole/funcs/<nom_fichier>.py.

Pratique

- Faire une fonction personnalisée qui retourner un texte ;
- créer une variable qui utilise cette fonction ;
- utiliser cette variable dans le template ircd.motd ;
- utiliser la fonction directement dans le template ircd.motd ;
- tester.

Redéfinition

- Permet de redéfinir des attributs et caractéristiques :
  - service
  - variable
- remove_check : supprimer les "check" déjà définit de la variable.

Pratique

- Redéfinir le texte de la variable web_url ;
- passer la variable activer_clam en mode expert.

Existance d'une variable

- Créer la variable si elle n'existe pas encore avec une valeur par défaut ;
- exemple : 'activer_clam' dans 23_proxy.xml ;
- attention la variable ne peut pas être créé après.

Script instance ou reconfigure

- Utilité :
  - tout ce qui n'est pas possible via les dictionnaires,
  - création de répertoire/changement de droit,
  - copie de fichier, suppression de fichier, déplacement de fichier,
  - création de base,
  - ... ;
- principe :
  - même script exécuter à l'instance et reconfigure,
  - ne doit pas poser de question au reconfigure !

Script instance ou reconfigure

- Emplacement des scripts :
  - /usr/share/eole/preservice/ : avant l'arrêt des services,
  - /usr/share/eole/pretemplate/ : avant la templatisation,
  - /usr/share/eole/posttemplate/ : entre la templatisation et redémarrage des services,
  - /usr/share/eole/postservice/ : après redémarrage des services ;
- compatible run-part :
  - pas d'extension,
  - exécutable,
  - script bash, python, ... il faut le préciser avec un shebang (#!/bin/bash),
  - premier argument : instance|reconfigure,
  - doivent toujours retourner le code de sortie "0" sauf si problème.

Écrire en couleur

- . /usr/lib/eole/ihm.sh
- EchoRouge
- EchoVert
- EchoOrange
- ...
- EchoGras

Question

- Question_ouinon
- $1 : contenu de la question de type oui/non
- $2 : interactif ou non
- $3 : valeur par défaut (défaut non)
- $4 : info|warn|err (défaut info)

Pratique

- EchoOrange 'couleur orange'
- EchoGras 'le titre'
- Question_ouinon "Voulez vous vraiment faire cette action ?"
- echo $?

Gestion des services

- CreoleService
  - $1 : nom du service
  - $2 : start|stop|restart|status
  - -c conteneur : pour un unique conteneur
- redémarrer tous les services : StartAll

Pratique

- CreoleService ntp stop
- CreoleService smbd stop
- CreoleService apache2 restart
- CreoleService apache2 restart -c web

Execution de commande

- CreoleRun
- $1 : commande
- $2 : conteneur

Execution de commande

- tcpcheck
- $1 : timeout
- $2 : ip :port

Pratique

- CreoleRun "echo mot" fichier
- test service : tcpcheck 2 192.0.2.52 :80

Gestion des locks

- CreoleLock acquire "nom_du_lock"
- CreoleLock release "nom_du_lock"
- CreoleLock is_locked "nom_du_lock"

Pratique

- Créé un lock "test" ;
- Supprimé le lock "test".

Valeur d'une variable

- CreoleGet nom_variable

Pratique

- Récupérer la valeur de la variable adresse_ip_eth0

Modifier la valeur

- CreoleSet nom_variable valeur ;
- Pour une liste : CreoleSet nom_variable """valeur1
- valeur2""" ;
- attention, ne peut changer la longueur de esclave.

Pratique

- Modifier la valeur de la variable de vm_swappiness à 10 ;
- modifier ubuntu_update_mirrors à "eole.ac-dijon.fr inconnu.lan"

Pratique

- Faire un script posttemplate qui créé le répertoire /tmp/repertoire ;
- si le répertoire existe, proposer de le supprimer à l'instance ;
- redémarrer le service "cron".

Création d'un script diagnose

- Diagnose sert à tester le service d'un serveur ;
- si possible, doit tester le bon fonctionnement de l'application ;
- script bash dans /usr/share/eole/diagnose.

Fonction diagnose standard

- Certains nombres de fonction dans /usr/lib/eole/diagnose.sh :
  - TestIP2 : test une IP via paquet ICMP ECHO_REQUEST ;
  - TestARP : requète ARP (évite les problèmes de firewall) ;
  - TestService : test une connexion TCP ;
  - TestUDP : vérifie l'écoute d'un service UDP ;
  - TestPid : test de la présence d'un processus via pidof ;
  - TestHTTPPage : test une page web particulière ;
- fonctions d'affichage :
  - EchoGras : titre de section ;
  - printf ". %$len_pfs => " "Test a afficher" ;
  - EchoVert|EchoRouge : succès ou échec ;
  - Inactif : si un service est inactif volontairement ;
  - NoConfig : service non configuré.

Pratique

- Faire un fichier diagnose pour IRCD.

API Python

Écrire en couleur

- from pyeole.ansiprint import *
- print_red
- print_green
- print_orange
- ...
- print_title

Question

- from pyeole.ihm import question_ouinon
- question_ouinon
  - question : contenu de la question de type oui/non
  - default : valeur par défaut (défaut non)
  - level : info|warn|err (défaut info)
  - return oui ou non

Pratique

- print_orange('couleur orange')
- print_title('le titre')
- a=question_ouinon('voulez vous vraiment faire cette action')

Gestion des services

- from pyeole.service import *
- manage_service :
  - action : start|stop|restart|status,
  - service : nom du service,
  - container : nom du conteneur (défaut root).

Pratique

- a=service_out('ntp', 'stop')
- a=service_code('smbd', 'stop', 'fichier')
- a=service_code_no_container('apache2', 'restart')

Execution de commande

- from pyeole.process import *
- system_code : exécution de commande, stdout et stderr dans la console
- system_out : rien dans la console
  - cmd : commande (list) ;
  - stdin : valeur de STDIN ;
  - container : nom du conteneur (root par défaut) ;
  - env : variable d'environnement ;
  - *context : ssh ou chroot (True par défaut) ;
  - *pty : créé un pseudo terminal (False par défaut).

Pratique

- a=system_code(['echo', 'mot'])
- a=system_out(['echo', 'mot'])
- stdin : a=system_code('cat', stdin="mot")
- env : a=system_code('env', env='mot' : 'mot')
- container : a=system_code(['ls', '/etc/samba/'], container='fichier')
- test service : tcpcheck('192.0.2.52', '139')

Gestion des locks

- from pyeole.lock import *
- acquire : ajout d'un lock
- release : supprimer un lock
- is_locked : vérifier la présence d'un fichier lock

Pratique

- ajouter, vérifier et supprimer un lock.

Lire valeur du dictionnaire

- Via creoled :
  - from creole.client import CreoleClient
  - client = CreoleClient()
  - client.get_creole(variable)
- en chargeant les dictionnaires :
  - from creole.loader import creole_loader
  - config = creole_loader()
  - config.creole.famille.variable
  - config.creole.famille.master.slave

Modifier valeur du dictionnaire

- from creole.loader import creole_loader, config_save_values
- config = creole_loader(rw=True)
- config.creole.famille.variable = value
- config.creole.famille.master.master.append(value)
- config.creole.famille.master.slave[-1] = value
- config_save_values(config)

Pratique

- Afficher la valeur de la variable adresse_ip_eth0 ;
- afficher la valeur de la variable netmask_ssh_eth0.

Schedule

- Créé pour éviter les conflicts entre mise à jour/sauvegarde ;
- permet de distinger les tâches avant sauvegarde/après sauvegarde ;
- principe :
  - extration avant sauvegarde,
  - suppression de fichier/mise à jour/ ... après la sauvegarde.

Schedule

- Script principalement bash ;
- enregistrer dans /usr/share/eole/schedule/scripts ;
- au format "run-part" ;
- dictionnaire d'activation dans /usr/share/eole/creole/extra/schedule/.

Pratique

- Faire un Schedule qui stop IRCD avant la sauvegarde ;
- faire un script qui démarre IRCD après la sauvegarde.

Console EAD

- Permet de rendre des scripts accessible dans l'EAD ;
- pas d'option possible ;
- dans l'EAD : Système/Console ;
- /usr/share/ead2/backend/actions/cmd_xxx.py : code de la commande ;
- /usr/share/ead2/backend/config/cmds/xxx.cmd : enregistrement de la commande ;
- /usr/share/ead2/backend/config/actions/actions_xxx.cfg : enregistrement de l'action ;
- /usr/share/ead2/backend/config/perms/perm_xxx.ini : association de rôle.

Pratique

- Faire un script avec "ls /tmp".

Fichier/répertoire dans la sauvegarde

- Prévoir les extractions ;
- fichier/répertoire spécifié dans /etc/bareos/bareosfichiers.d/.

Plan

Cadoles

Personnalisation d'un serveur

- Liste des fichiers sauvegardés : dans la page d'état du serveur cliquer sur "voir les fichiers personnalisés" ;
- pour les dictionnaires locaux :
  - dans la partie "module" du Zéphir, cliquer sur "Dictionnaires personnalisés",
  - activer les modules dans la page "voir les fichiers personnalisés" de la page d'état.

Pratique

- Ajouter les fichiers personnalisés dans l'application Zéphir pour Scribe ;
- ajouter la dépendance pour les paquets ;
- envoyer la configuration.

Variante

- Configuration identique applicable à plusieurs serveurs ;
- par défaut chaque module est dans la variante "standard" ;
- création des templates, patchs et dicos éventuels ;
- déplacement dans /usr/share/eole/creole/patch/variante et /usr/share/eole/creole/dicos/variante ;
- fichiers ou paquets pas référencés : /usr/share/zephir/zephir_conf/fichiers_variante ;
- dans /usr/share/zephir/scripts : ./creation_variante ;
- mettre à jour une variante : ./maj_variante.

Pratique

- Créer une variante ;
- mettre le serveur dans la variante ;
- enregistrer le serveur dans Zéphir.

Plan

Synchronisation

- Il est possible de se connecter sur gen_config avec un compte Zéphir ;
- différence entre fichier local et distant.

Plan

Cadoles

Exécuter un script Zéphir

- Exécution sur un serveur ou un groupe de serveurs ;
- les scripts doivent être dans /usr/share/zephir/scripts/ ;
- ils doivent avoir l'extension .zephir ;
- ne pas spécifier l'extension lors de l'exécuxion ;
- fonctionseole.init_proc : permet de vérifier si l'action est bloqué ;
- fonctionseole.zephir(<etat>, <message>, <type>) : log envoyer au Zéphir ;
- les scripts sont lancés en tant que uucp, si besoin utiliser sudo_script('monscript.zephir').

Pratique

- Créer un script personnalisé permettant un supprimer un fichier dont le nom est en paramètre.

Plan

Préparation de la variante

- Il faut créer une variante "équivalente" sur la nouvelle version ;
- pour les modules compatibles entre version, un bouton "Import des données x.x.x" est disponible :
  - dictionnaires locaux copiés,
  - variante copiée et déclarée équivalente,
  - valeurs par défaut copiées,
  - possibilité de relance ultérieur (copie les nouveaux dictionnaires) ;

Pratique

- Faire une variante dans une version inférieure ;
- migrer la variante dans la nouvelle version.

Migration de la configuration depuis Zéphir

- Préparer à l'avance la configuration des serveurs à migrer ;
- met à jour les informations en concervant l'identifiant ;
- dictionnaire/template/patch personnalisée sont à migrer à la main ;
- sur la page d'état d'un serveur : 'générer les données de migration' ;
- choisir la variante à utiliser ;
- formulaire de saisie ;
- "Sauver sur Zéphir".

Page de suivi de la migration

- Sur la page d'accueil "suivi de la migration" ;

Migration avec réinstallation

- Seule méthode possible pour Horus/Scribe 2.2 vers version supérieure ;
- pas de migration automatique (problème mise à jour Ubuntu) ;
- outils externes :
  ftp ://eoleng.ac-dijon.fr/pub/Outils/migration/ :
  migration2X.sh ;
- nécessite un disque distant/usb/local en ext3 ;
- installer le module 2.X ;
- conversion du fichier config.eol ;
- instance du serveur ;
- relancer le script.

Migration sans réinstallation

- Utilisation du script Upgrade-Auto ;
- migration du serveur dans le Zéphir en fin de procédure.

Migration Horus/Scribe 2.X.Y vers une release supérieure

- Utilisation du script Maj-Release
-
-
-
-

Licence