

Mon petit projet libre ... ne connaît pas la crise

Auteur : Carl Chenet

Durée : 40 minutes

Naissance du projet Belier

<http://www.ohmytux.com/belier>

Le besoin et l'existant

- Administrer un parc de 2000 serveurs

Le besoin et l'existant

- Administrer un parc de 2000 serveurs
- Pas d'administration centralisée

Le besoin et l'existant

- Administrer un parc de 2000 serveurs
- Pas d'administration centralisée
- Trop d'équipes projets

Le besoin et l'existant

- Administrer un parc de 2000 serveurs
- Pas d'administration centralisée
- Trop d'équipes projets
- Le parc bouge tout le temps

Un script ne suffit plus

- Gérer les cas spéciaux

Un script ne suffit plus

- Gérer les cas spéciaux
- Comportement généralisable

Le choix du langage

- Vous êtes le développeur

Le choix du langage

- Vous êtes le développeur
- Certains langages s'imposent

Le choix du langage

- Vous êtes le développeur
- Certains langages s'imposent
 - Besoin de perfs : langage C

Pourquoi Python ?

- Python présent en installation minimale

Pourquoi Python ?

- Python présent en installation minimale
- Robustesse de la bibliothèque de base

Pourquoi Python ?

- Python présent en installation minimale
- Robustesse de la bibliothèque de base
- Portable

L'architecture logicielle

- Quel niveau de flexibilité

L'architecture logicielle

- Quel niveau de flexibilité
- KISS (Keep It Simple Stupid!)

L'architecture logicielle

- Quel niveau de flexibilité
- KISS (Keep It Simple Stupid!)
- Faire une seule tâche, et bien la faire

L'internationalisation 1/2

- Dès les premières lignes de votre programme

L'internationalisation 1/2

- Dès les premières lignes de votre programme
- Plus tard, perte de temps et travail ennuyeux

L'internationalisation 1/2

- Dès les premières lignes de votre programme
- Plus tard, perte de temps et travail ennuyeux
- Facile à mettre en place en Python

L'internationalisation 2/2

- Module gettext

L'internationalisation 2/2

- Module gettext
- Entourer vos chaînes avec `_()`

L'internationalisation 2/2

- Module gettext
- Entourer vos chaînes avec `_()`
- Générer les chaînes à traduire avec `xgettext`

L'internationalisation 2/2

- Module gettext
- Entourer vos chaînes avec `_()`
- Générer les chaînes à traduire avec `xgettext`
- Générer le fichier `.mo` avec `msgfmt`

L'internationalisation 2/2

- Module gettext
- Entourer vos chaînes avec `_()`
- Générer les chaînes à traduire avec `xgettext`
- Générer le fichier `.mo` avec `msgfmt`
- Déclarer `gettext.install()`

Le projet prend de l'ampleur

Gestionnaire de sources 1/2

- Historique des modifications

Gestionnaire de sources 1/2

- Historique des modifications
- Commentaires associés aux modifications

Gestionnaire de sources 1/2

- Historique des modifications
- Commentaires associés aux modifications
- On sait qui aller voir

Gestionnaire de sources 1/2

- Historique des modifications
- Commentaires associés aux modifications
- On sait qui aller voir
- Accès permanent à toutes les versions

Gestionnaire de sources 2/2

- Décentralisé

Gestionnaire de sources 2/2

- Décentralisé
- Performant

Gestionnaire de sources 2/2

- Décentralisé
- Performant
- Modifiable (argument de GvR pour Mercurial)

Gestionnaire de sources 2/2

- Décentralisé
- Performant
- Modifiable
- Git, Mercurial

Git

- Git, les branches !

Git

- Git, les branches !
- Lien entre git et l'empaquetage

Évaluer la qualité du code

- Détecter les erreurs sans exécuter le code

Évaluer la qualité du code

- Détecter les erreurs sans exécuter le code
- Code conforme aux standards

Évaluer la qualité du code

- Détecter les erreurs sans exécuter le code
- Code conforme aux standards
 - Code conforme à vos standards

Évaluer la qualité du code

- Détecter les erreurs sans exécuter le code
- Code conforme aux standards
 - Code conforme à vos standards
- Détection des doublons dans le code

Évaluer la qualité du code

- Détecter les erreurs sans exécuter le code
- Code conforme aux standards
 - Code conforme à vos standards
- Détection des doublons dans le code
- Détection des potentielles failles de sécurité

Style du code et erreurs

- Pylint : style et erreurs

Style du code et erreurs

- Pylint : style et erreurs
 - Par défaut suit les standards de style Python

Style du code et erreurs

- Pylint : style et erreurs
 - Par défaut suit les standards de style Python
 - Très configurable

Style du code et erreurs

- Pylint : style et erreurs
 - Par défaut suit les standards de style Python
 - Très configurable
- Pyflakes : erreurs

Style du code et erreurs

- Pylint : style et erreurs
 - Par défaut suit les standards de style Python
 - Très configurable
- Pyflakes : erreurs
- Pychecker : erreurs

Doublons dans le code

- CloneDigger

Doublons dans le code

- CloneDigger
- Réduire la maintenance

Doublons dans le code

- CloneDigger
- Réduire la maintenance
- Réduire le nombre d'erreurs possibles

Analyse des risques de sécurité

- RATS : Rough Auditing Tool for Security

Analyse des risques de sécurité

- RATS : Rough Auditing Tool for Security
- C, C++, Perl, PHP et Python

Les tests unitaires

- La classe ou la fonction fait-elle son travail ?

Les tests unitaires

- La classe ou la fonction fait-elle son travail ?
- Éviter les régressions

Les tests unitaires

- La classe ou la fonction fait-elle son travail ?
- Éviter les régressions
 - Ça marchait avant mais ça ne marche plus

Les tests unitaires

- La classe ou la fonction fait-elle son travail ?
- Éviter les régressions
 - Ça marchait avant mais ça ne marche plus
 - RÉGRESSION!

Le module unittest

- Écriture de tests unitaires

Générer la documentation

- Avoir une documentation dans le code

Générer la documentation

- Avoir une documentation dans le code
- Faire évoluer le code et la documentation

Générer la documentation

- Avoir une documentation dans le code
- Faire évoluer le code et la documentation
- pydoc

Le module doctest

- Validité des exemples de vos commentaires

Le module doctest

- Validité des exemples de vos commentaires
- Très utile pour une API

Le module doctest

- Validité des exemples de vos commentaires
- Très utile pour une API
- Au sein du module ou dans un fichier externe

Le fuzzing

- Test automatisé de l'application

Le fuzzing

- Test automatisé de l'application
- Du point de vue de l'utilisateur

Le fuzzing

- Test automatisé de l'application
- Du point de vue de l'utilisateur
- On envoie les données sur les entrées

Le fuzzing

- Test automatisé de l'application
- Du point de vue de l'utilisateur
- On envoie les données sur les entrées
 - Entrée standard

Le fuzzing

- Test automatisé de l'application
- Du point de vue de l'utilisateur
- On envoie les données sur les entrées
 - Entrée standard
 - Fichiers

Le fuzzing

- Test automatisé de l'application
- Du point de vue de l'utilisateur
- On envoie les données sur les entrées
 - Entrée standard
 - Fichiers
 - Port réseau

Le fuzzing

- Test automatisé de l'application
- Du point de vue de l'utilisateur
- On envoie les données sur les entrées
 - Entrée standard
 - Fichiers
 - Port réseau
- On enregistre le déroulement des sessions qui échouent

Fusil

- Bibliothèque de fuzzing

Fusil

- Bibliothèque de fuzzing
- En Python

Fusil

- Bibliothèque de fuzzing
- En Python
- Upstream très réactive

Compilation continue

- Contrôle régulier du cycle compilation/test

Compilation continue

- Contrôle régulier du cycle compilation/test
- Rapport de tests clair

Compilation continue

- Contrôle régulier du cycle compilation/test
- Rapport de tests clair
- Buildbot

La version stable

Un dépôt public pour les sources

- Rendre disponible l'application

Un dépôt public pour les sources

- Rendre disponible l'application
- Avoir des retours

Un dépôt public pour les sources

- Rendre disponible l'application
- Avoir des retours
 - Nécessite une infrastructure

Un dépôt public pour les sources

- Rendre disponible l'application
- Avoir des retours
 - Nécessite une infrastructure
 - Mailing-list

Un dépôt public pour les sources

- Rendre disponible l'application
- Avoir des retours
 - Nécessite une infrastructure
 - Mailing-list
 - Forum

Un dépôt public pour les sources

- Rendre disponible l'application
- Avoir des retours
 - Nécessite une infrastructure
 - Mailing-list
 - Forum
- Commencer à se faire connaître

Automatiser l'installation

- Ménager la 1ère impression

Automatiser l'installation

- Ménager la 1ère impression
- Ne pas décourager

Automatiser l'installation

- Ménager la 1ère impression
- Ne pas décourager
- Vous facilite la vie

Automatiser l'installation

- Ménager la 1ère impression
- Ne pas décourager
- Vous facilite la vie
 - Documentation moins complexe

Automatiser l'installation

- Ménager la 1ère impression
- Ne pas décourager
- Vous facilite la vie
 - Documentation moins complexe
 - Support utilisateur moins important

Indépendantes des distributions

- Distutils

Indépendantes des distributions

- Distutils
 - Python setup.py install

Indépendantes des distributions

- Distutils
 - Python setup.py install
- Setuptools et PyPI

Indépendantes des distributions

- Distutils
 - Python setup.py install
- Setuptools et PyPI
 - Python setup.py register sdist upload

Indépendantes des distributions

- Distutils
 - Python setup.py install
- Setuptools et PyPI
 - Python setup.py register sdist upload
 - easy_install

Spécifiques aux distributions

- Redhat :

Spécifiques aux distributions

- Redhat :
 - Distutils permet de générer un .rpm

Spécifiques aux distributions

- Redhat :
 - Distutils permet de générer un .rpm
- Debian :

Spécifiques aux distributions

- Redhat :
 - Distutils permet de générer un .rpm
- Debian :
 - CDBS

Spécifiques aux distributions

- Redhat :
 - Distutils permet de générer un .rpm
- Debian :
 - CDBS
 - git-buildpackage

Faire connaître son projet 1/3

- Le local avant l'international

Faire connaître son projet 1/3

- Le local avant l'international
 - Site web communautaire

Faire connaître son projet 1/3

- Le local avant l'international
 - Site web communautaire
 - Forums logiciels libres

Faire connaître son projet 1/3

- Le local avant l'international
 - Site web communautaire
 - Forums logiciels libres
 - Vos collègues

Faire connaître son projet 2/3

- L'artillerie :
 - Freshmeat

Faire connaître son projet 2/3

- L'artillerie :
 - Freshmeat
 - Sourceforge

Faire connaître son projet 2/3

- L'artillerie :
 - Freshmeat
 - Sourceforge
 - Ohloh

Faire connaître son projet 3/3

- Intégrer le logiciel dans différentes distributions
 - Retour des empaqueteurs

Faire connaître son projet 3/3

- Intégrer le logiciel dans différentes distributions
 - Retour des empaqueteurs
 - Disponibilité de l'application

Livre

- Expert Python programming , Tarek Ziadé

Merci de votre attention

ARTEMYS

www.artemys.fr

Liens

- Git : <http://git-scm.com/>
- Pylint : <http://www.logilab.org/857>
- Pyflakes : <http://www.divmod.org/trac/wiki/DivmodPyflakes>
- Pychecker : <http://pychecker.sourceforge.net/>
- CloneDigger : <http://clonedigger.sourceforge.net/>
- RATS : <http://www.fortify.com/security-resources/rats.jsp>
- Module unittest : <http://www.python.org/doc/current/library/unittest.html#module-unittest>
- Module doctest : <http://www.python.org/doc/current/library/doctest.html#module-doctest>
- Fusil : <http://fusil.hachoir.org/>
- Buildbot : <http://buildbot.net/trac>
- Distutils : <http://docs.python.org/library/distutils.html>
- Setuptools : <http://peak.telecommunity.com/DevCenter/setuptools>