

オープンソースによる システム管理の自動化

“Real Sysadmins Don't Login”

Tomoyuki Sakurai
<trombik@gentoo.gr.jp>

Some questions...

最近、OSをインストールしましたか？

インストール作業は自動化されていきましたか？

設定作業は自動化されていきましたか？

設定が変更されていないことを保証できますか？

誰が何を変更したかわかりますか？

Before configuration management

- ラックにサーバを設置する
- ネットワークに接続する
- OSをインストールする
- OSをアップデートする
- 必要なソフトウェアをインストールする
- ソフトウェアを設定する

After configuration management

- ラックにサーバを設置する
- ネットワークに接続する
- しばらく待つ

behind the scene...

- pxebootとsysinstall/kickstartがパッチ適用済みのOSをインストール
- reboot後にcfengineを起動
- cfagentが設定とパッケージのインストール
- 何度でも再現可能

cfengine

Before cfengine

- 1台ごとにコンソールでインストール
- インストール後の設定はばらばら
 - 実際の設定と手順書の乖離
 - ベースラインは絵に描いた餅
- 誰が何をしたのかがわからない
 - 「何も変更していないはずです」

After cfengine

インストールから自動化

- 作業者のスキルに依存せず、同じ環境を忠実に再現
- ベースラインを共通化し、役割に応じて個別に設定
- 変更されていないことの保障

設定を抽象化し、混在環境でも同様に管理

- OS-based
- role-based

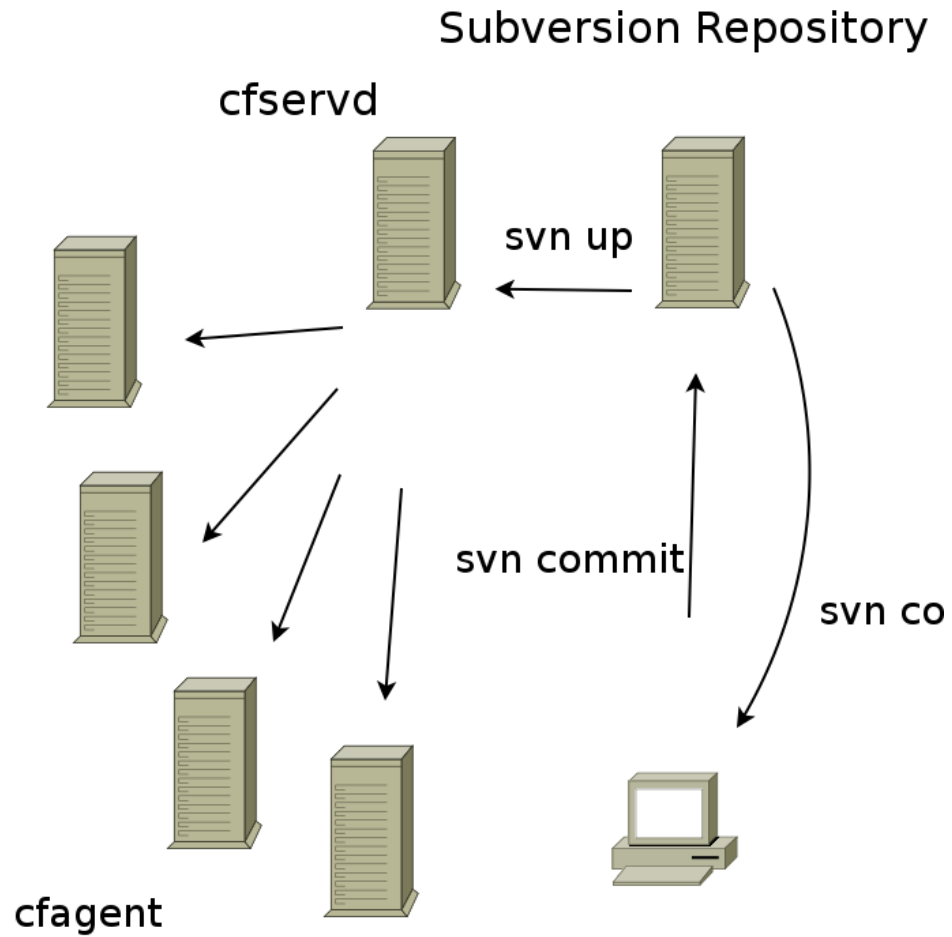
変更の監査とroleback

- CVS/Subversion

cfengineとは

- Configuration Management System
- 設定の一元管理
- 「管理パターン」
 - ファイルのコピー/削除/編集 (/etc/hosts)
 - daemonの管理
 - パッケージ管理 (rpm/portage/dpkgのみ)

Overview



cfengineはすばらしかった

- 設定を一括で管理できた
- 手順のバックアップができた
- あるべき姿に維持されていることを保障できた
- 使い捨てのshやPerlスクリプトを追放できた
- 管理作業のパターン化により品質がある程度均一化された
- 属人的な管理作業から開放された

cfengineはすばらしかった(cont.)

- でも、進歩がなさ杉
- 依存関係や関連性を持たせられない
 - 正しい順序を指定するのは至難の技
- Classの継承ができない
- cfengineでできないことは、結局shを書いて解決するはめに
- 「cfengineの設計はいいが、実装がダメ」

だめじゃん

Puppet

... is so sexy

The next generation tool for sysadmin

Puppet to the rescue!

- なんでも抽象化
 - ユーザの追加は `adduser?useradd?pw?`
 - パッケージのアップデートは `ports?yum?emerge?`
- 拡張性
 - 主要なパッケージマネージメントシステムをサポート
 - `sh`によるhackにさようなら
- `written by a real sysadmin in Ruby`
 - きれいなコード
 - そのうち30%はテスト

Cfengine is not sexy, puppet is

- 高い拡張性
 - 複数のリソースを1つのclassとして定義できる
 - classのinheritも可能
 - 足りない”type”は定義してしまえ
- 依存関係も処理できまっせ
 - httpd.confを更新したらhttpdを再起動したいでしよう?
- LDAPサポート (!)
 - Node情報をLDAPツリーに

Example site.pp

```
$default_owner = $operatingsystem ? {  
  default => root  
}  
$default_group = $operatingsystem ? {  
  FreeBSD => wheel,  
  default => root  
}  
file { '/etc/hosts':  
  owner    => $default_owner,  
  group    => $default_group,  
  mode     => 644,  
  source   => 'puppet://server/module/hosts',  
}
```

Case文っぽく分岐して
変数に代入できます

ライブラリ (factor) が
適切な値やモジュールを
判定します

/etc/hostsを
一元管理

これはすべてのホストに
適用されます

Example site.pp (cont.)

```
class sudo {
  $sudo = $operatingsystem ? {
    FreeBSD => 'security/sudo',
    default => sudo,
  }
  $sudoers = $operatingsystem ? {
    FreeBSD => '/usr/local/etc/sudoers',
    default => '/etc/sudoers'
  }
  package { $sudo:
    ensure => latest,
    alias  => sudo
  }
  file { $sudoers:
    source  => 'puppet://server/module/sudoers',
    mode    => 440,
    owner   => $default_owner,
    group   => $default_group,
    alias   => sudoers
  }
}
node www { include sudo }
node mta { include sudo }
```

複数のリソースをclass
として定義できます

依存関係

```
Class ssh {
  package { 'ssh':
    ensure => latest,
    name    => $operationsystem ? {
      FreeBSD => 'security/openssh',
      default  => 'openssh'
    },
  },
  file { '/etc/ssh':
    source    => puppet://server/module/ssh,
    recurse  => true,
    notify   => Service[ssh]
  },
  service { 'ssh':
    name      => sshd,
    ensure    => running,
    subscribe => Package[ssh]
  },
}
node www { include ssh }
```

各リソースは互いに関連して
ひとつのサービスを構成します

ログと監査

- CVS/Subversion/svk...
 - blame/diff/log
 - hookでML/IRC bot/RSSに出力
 - Web UI? つSVN::Web/trac
- stagingだって、テスト環境と本番環境をmergeするだけ
- とりあえずリポジトリに入れてしまえばこっちのもの

Configuration Management Systemと セキュリティ

ポリシーの強制

- たくさんのHowToやTipsがあります
- 問題は、それをどうやって設定し、維持していくか

「セキュリティポリシーが一部正しく遵守されていない状態」

「はてなサーバーへの不正な侵入について」[1]

admin⁸ blog configuration deferred devel
fedora⁶ fedora core howto infra
iptables¹⁵ linux⁸⁴ manage network os
root secure security⁶⁴ server⁷² setting
setup share sudo tips³⁰ tmp unix¹⁶ webdev
webシステム² あとで読む² あとで読んだら消す お役立ち
これは便利 ほお ウェブデザイン ウノウ サーバ³⁸
サーバー¹⁴ サーバ関連 サーバ構築³
サーバ構築の方法 サーバ管理¹⁴ システム管理
セキュリティ⁴⁷ ノウハウ プログラム 基本 情報
誌・情報源 殿堂入り物件 目から鱗 知識 管理⁹ 設定
読み物 運用³⁰ 開発

改ざん検知

- うっかりミス対策
- 変更してもリポジトリの設定が優先
- 悪意のある改ざんには専用のソフトウェアを
 - ◆ Osiris, Samhain, AIDE...
 - ◆ A comparison of several host/file integrity checkers (scanners) [2]

侵入後の復旧

- バックアップだけでは役に立ちません
- 復旧はOSの再インストールが基本
- 「手順書に基づいて再構築します」
 - ◆ 手動で?
 - ◆ 文書化されていない手順?
 - ◆ 「最終更新日 1999/m/d」
 - ◆ 同じ設定が再現できることを保証できますか?

Puppetの現状と今後

- 開発は活発
- 各distribution側でもサポートが進む
 - ◆ FreeBSD (official sysutils/puppet)
 - ◆ Gentoo (in bugs)
 - ◆ OpenBSD (official)
 - ◆ Debian (official)
 - ◆ SuSE (official)
 - ◆ Fedora (official?)
- 採用事例も増加中
 - ◆ Organizations using Puppet [3]

RANCID

「こじつけすぎだろ」

Really Awesome New Cisco conflg Differ

L2/L3の世界

- いまだにtelnetが幅を利かせる
- 設定ファイルはtftpで
- コンソールかGUIの2択
- 付属品か高価なツールの2択
 - 高価で使えないツールだと目も当てられない
- ベンダー依存
- 貧弱なログ

ハックの余地は多くない

L2/L3スイッチだってリビジョン管理

- Perl+expect+telnet/SSHでshow conf
- 変更があればCVSにcommit、差分はメールで
 - CVSにcommitしてconfigに反映はできない
- 複数のスイッチでコマンドをバッチ処理
- マルチベンダサポート
 - Cisco/Extreme/Foundry/NetScreen...
- 正直、hackであることは否めない

But, it works!

- コスト $\doteq 0$
- バックアップ with 履歴
 - Commit logは残らないけどね
- マルチベンダでも平気
- master/slaveのdiff
- 複数のスイッチにまとめてconfig投入
- 1か所に集めたconfigをhack

Conclusion

- 自動化はミスが入る余地を減らします
- 再発明した壊れた車輪をフレームワークで置き換えると、品質の底上げと均一化をもたらします
- 再現が保障されたシステムは、セキュリティ向上の一助になります
- 方法がきれいでなくても、リポジトリに入れる価値はあります

Question?

Resources

- Cfengine
<http://www.cfengine.org/>
- Reductive Labs / Puppet
<http://reductivelabs.com/projects/puppet/>
- RANCID
<http://www.shrubbery.net/rancid/>
- Puppetの作者、Luke A. Kanies氏によるO'Reillyの記事
<http://www.oreillynet.com/pub/au/237>
- [1] はてなサーバーへの不正な侵入について
<http://hatena.g.hatena.ne.jp/hatena/20070314/1173858953>
- [2] A comparison of several host/file integrity checkers
<http://www.la-samhna.de/library/scanners.html>
- [3] Organizations using Puppet
<http://reductivelabs.com/trac/puppet/wiki/WhosUsingPuppet>

GentooJPはミラーを探しています

- `/usr/portage/distfiles`
- 容量は約50GB
- 通信量は1.4TB/month 50GB/day
- HTTP/FTP
- くわしくは明日のGentooJP展示ブースで
 - ◆ `<trombik@gentoo.org>`
 - ◆ `irc://irc.freenode.net #gentoo-ja`