CFEngine 3

Nástroj pro hromadnou správu

Úvod

- Marek Petko
  - Student FAV na ZČU v Plzni
  - Distribuované systémy a počítačové sítě
- Hromadná správa výpočetních systémů v heterogenním prostředí
  - Diplomová práce pro CIV
  - Michal Švamberg (vedoucí práce)

Motivace

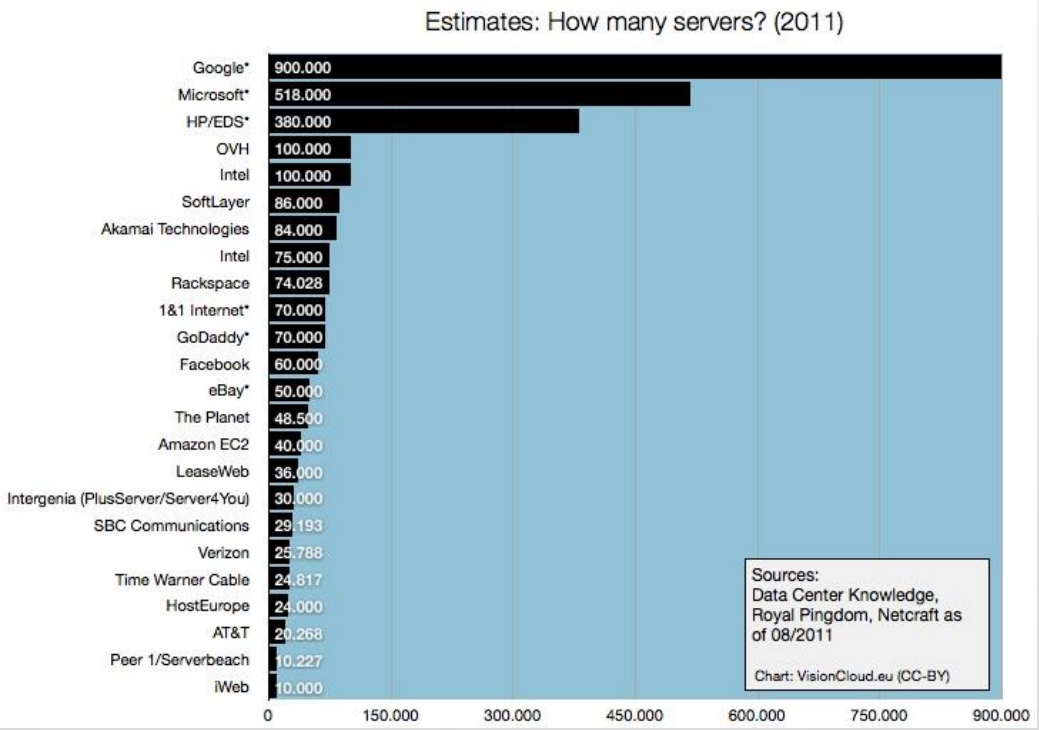
- Rozsáhlá datacentra
- Clustery
- Cloudy

Počet adminů

vs.

Počet serverů

Přímá úměrnost?

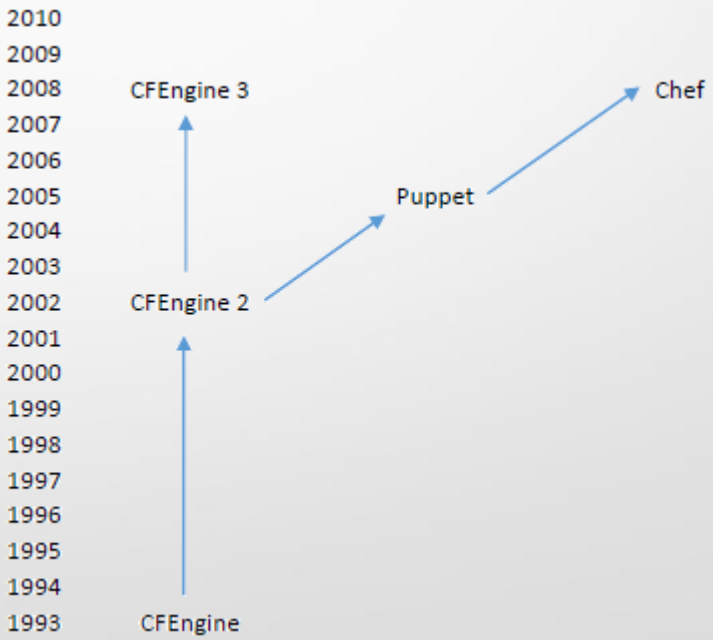


Možnosti

- CFEngine
- Puppet
- Chef

- Open Source
- Enterprise

- Další
  - Ansible
  - SaltStack



Placené verze

Servery	Počet	CFEngine 1 rok	CFEngine 3 roky	CFEngine 1 rok Education	CFEngine 3 roky Education	Puppet 1 rok	Chef 1 rok
GNU/Linux Debian	250	482 500 Kč	335 000 Kč	241 250 Kč	167 500 Kč	502 500 Kč	323 000 Kč
Windows Server	25	48 250 Kč	33 500 Kč	24 125 Kč	16 750 Kč	50 250 Kč	32 300 Kč
Oracle Solaris	11	21 230 Kč	14 740 Kč	10 615 Kč	7 370 Kč	22 110 Kč	14 212 Kč
Celkem	286	551 980 Kč	383 240 Kč	275 990 Kč	191 620 Kč	574 860 Kč	369 512 Kč
1 uzel	1	1 930 Kč	1 340 Kč	965 Kč	670 Kč	2 010 Kč	1 292 Kč

CFEngine 3

- Silný teoretický základ
  - Teorie slibů (Mark Burgess)
- Vývoj
  - Důraz na efektivitu (Kompilované C)
  - Důraz na bezpečnost
- != CFEngine 2

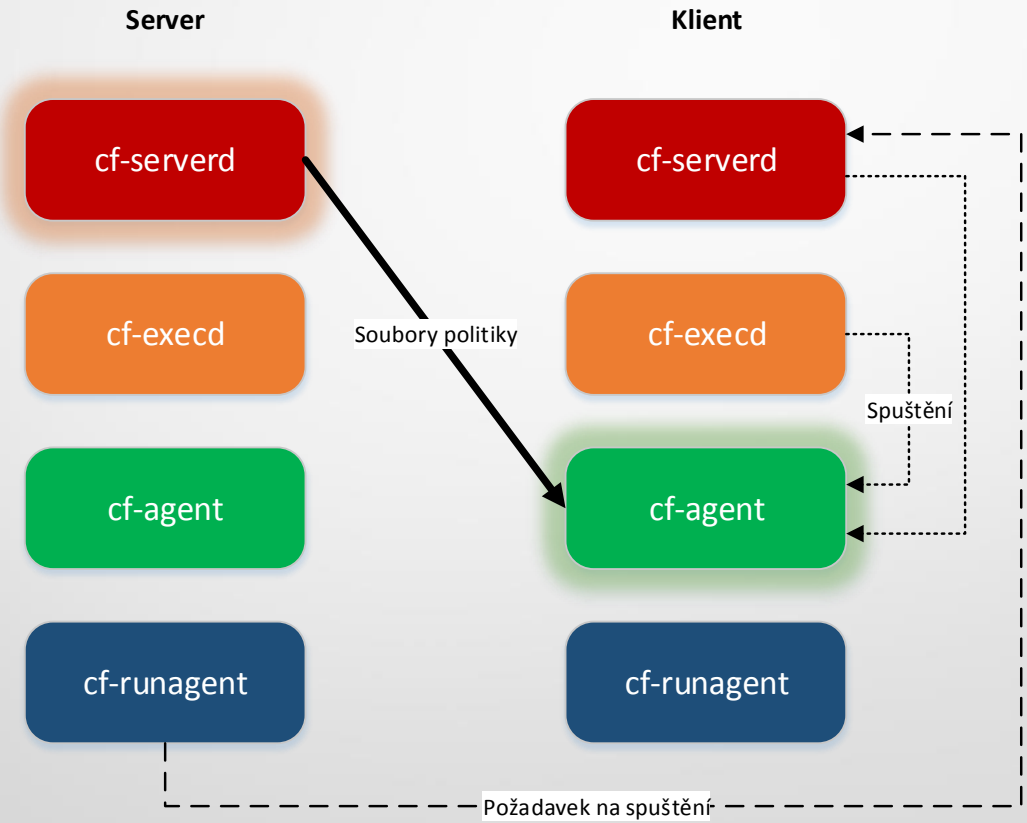
Princip funkce

- Jako orchestr
  - Každý obdrží noty a podle nich hraje
  - Dirigent poskytne noty, ale netahá houslistovi za smyčec
- Noty = konfigurace tzv. politika
- Configuration as code
- Kontrast k Ansible

Distribuce politiky (1)

- Klient – server
  - Pouze pull
  - Možno force pull
  - Nikdy push (bezpečnost)
- Politika
  - Se zpracovává pouze lokálně, autonomně
  - Na serveru plaintext – na klientech plaintext
  - Ze serveru se pouze kopíruje (ale nemusí!)
  - Není to skript

Distribuce politiky (2)



Bezpečnost

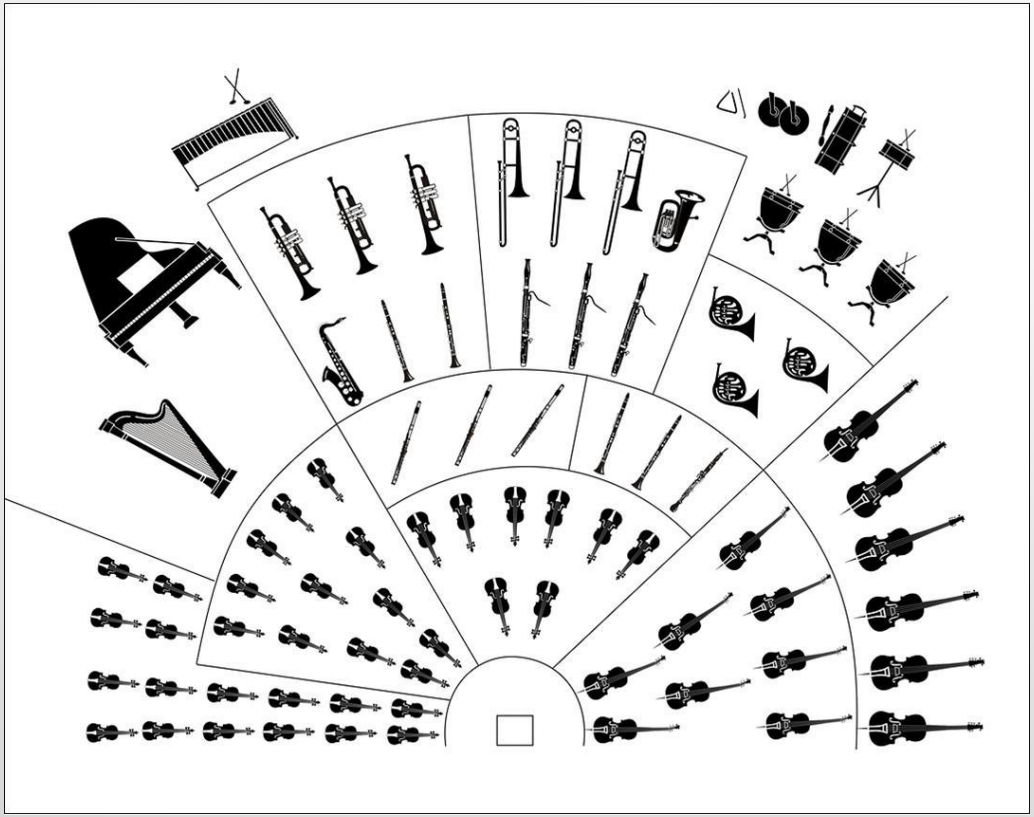
2.2 The principles of CFEngine security

CFEngine adheres to the following design principles:

1. It shall be, by design, impossible to send policy-altering data to a CFEngine agent. Each host shall retain its right to veto policy suggestions at all times. This is called the **Voluntary Cooperation Model**.
2. CFEngine will support the encryption of data transmitted over the network.
3. Each host shall continue to function, as far as possible, without the need for communication with other hosts.
4. CFEngine will use a lightweight peer model for key trust (like the Secure Shell). No centralized certficate authority shall be used. SSL and TLS shall not be used.
5. CFEngine shall always provide safe defaults, that grant no access to other hosts.

http://cfengine.com/manuals_files/SpecialTopic_Security.pdf

CFEngine prakticky



Instalace

- Community Edition
  - Přeložení zdrojového kódu (C99 + knihovny)
  - Instalace z balíku od CFEngine
  - Instalace z repositáře OS (!)
  - < 40MB
- Enterprise Edition
  - Uzavřený zdrojový kód
  - Jen balíky
  - Verze Free 25 Node

Souborová struktura

`/var/cfengine/`

- **`bin`** – binární spustitelné soubory
- **`inputs`** – politika k lokálnímu spuštění cf-agentem
- **`lastseen`** – "Log data for incoming and outgoing connections."
- **`lib`** – knihovny (=! standardní knihovna)
- **`masterfiles`** – repositář politik na serveru
- **`modules`** – složka pro moduly
- **`outputs`** – výstupy cf-agenta (jen `reports:`)
- **`ppkeys`** – klíče
- **`reports`** – seznamy opravených promise (jen Enterprise)
- **`share`** – dokumentace, příklady, zdrojové kódy, atd.
- **`state`** – stavová databáze

Koncepce jazyka

- Deklarativní jazyk
- Teorie slibů – Mark Burgess
- Cokoliv je slib (promise)

Bundle

- Je kolekce jednotlivých slibů
- Uvnitř dále členěno podle typu

- Bundle ~ subrutina, procedura…
  - Promise type
    - Class
      - Promise

Ukázkový bundle


```
bundle agent ntp
{
    files:

     "/etc/ntp.conf"
       create => "true",
       copy_from => secure_cp("/repo/config-files/ntp.conf",
                              "daidalos.civ.zcu.cz");

    services:

     "ntp"
       service_policy => "start";
}
```

Třídy

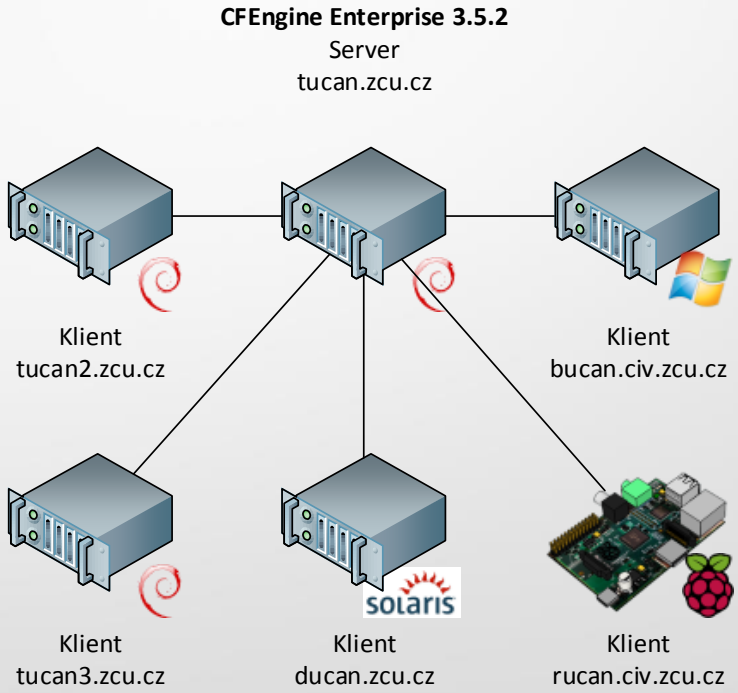
```
bundle agent hard_class
{
  reports:

    linux::
      "Tento klient používá Linux!";

    solaris::
      "Tento klient používá Solaris!";

    windows::
      "Tento klient používá Windows!";
}
```

```
bundle agent firewall
{
  vars:

    "iptables_output"
       string => execresult(/sbin/iptables -L INPUT, "noshell");

  classes:

    "firewall_disabled"
       expression => regcmp("(?ms).*^Chain INPUT \(policy ACCEPT\)$.*",
                            $(iptables_output));

    "shorewall"
       expression => fileexists("/etc/shorewall");

  commands:

    firewall_disabled.shorewall::

     "/etc/init.d/shorewall start"
        classes => if_repaired("firewall_enabled");

  reports:

    firewall_disabled.!firewall_enabled::

     "Firewall byl deaktivovan a nepodarilo se jej nastartovat.";
}
```

Nasazení na ZČU

Testovací prostředí

- Lab 1
  - Community Edition
  - 4ks Debian
- Lab 2
  - Enterprise Edition
  - Free 25 Node
  - 3ks Debian
  - 1ks Solaris
  - 1ks Windows
  - 1ks Raspbian na Raspberry



Produkční prostředí

- CFEngine Community 3.5.2
- Debian servery
- 36 klientů++
- Všechny nově instalované Debian servery přes FAI

Správa verzí politiky

- Udržování kompletní historie úprav
- Víceuživatelský přístup
- Vytváření vlastních větví konfigurace

↳ GIT na AFS

Větve konfigurace

- Master (testing)
- Production

- Přiřazení strojů v def.cf
  - Master – vyjmenovány
  - Production – vše ostatní

- Při aktualizaci politiky se stáhne příslušná větev

```
classes:
  "branch_master"
     or => {
      "cicomexocitl_civ_zcu_cz",
      "daphne_civ_zcu_cz",
      "metalist_civ_zcu_cz",
      "kiosek_tv10",
     };
```

Proces změn

1. Lokální klon repositáře z AFS
2. Nová lokální větev
3. Úpravy a testování na lokálním PC
4. Merge a push do testovací větve v repositáři
5. Kontrola na testovacích strojích
6. Merge a push do produkční větve

Rozšíření změn

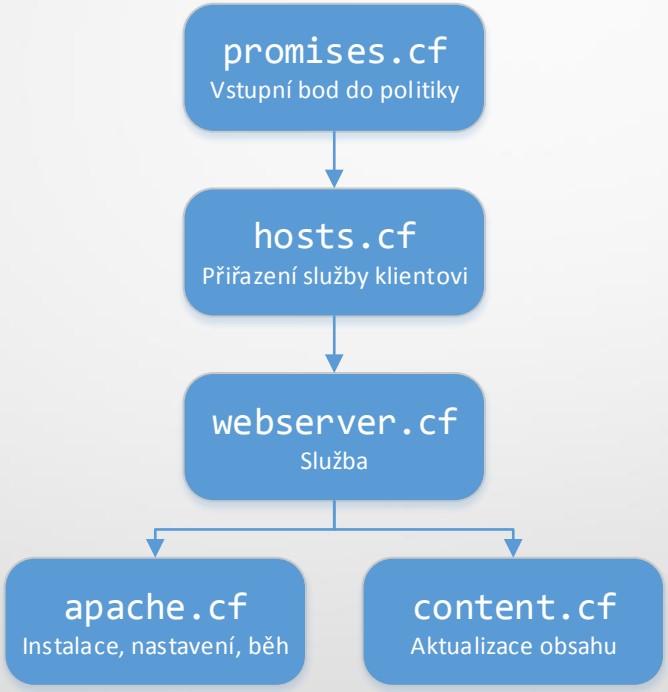
- Automaticky
- CFEngine na serveru spustí modul pro kontrolu GIT repositáře
  - Každých 5 minut→ git pull
  - Pozor na cf_promises_validated

Organizace politiky (1)

- **`controls/`** – konfigurace komponent
- **`inventory/`** – průzkum systému a vytváření nových tříd
- **`lib/`** – standardní knihovna
- **`modules/`** – moduly
- **`templates/`** – templaty ke kopírování na klienty
- **`def.cf`** – nastavení proměnných pro konfiguraci
- **`promises.cf`** – vstupní bod politiky
- **`update.cf`** – proces aktualizace politik na klientech ☠ ☠ ☠

- **`services/`** – vlastní politiky podle service oriented přístupu
- **`files/`** – soubory ke kopírování na klienty
- **`zcuinventory/`** – průzkum systému a vytváření nových tříd
- **`zculib/`** – vlastní knihovna
- **`hosts.cf`** – spouštění service bundlů podle hostů nebo skupin

Organizace politiky (2)



Zpětná vazba

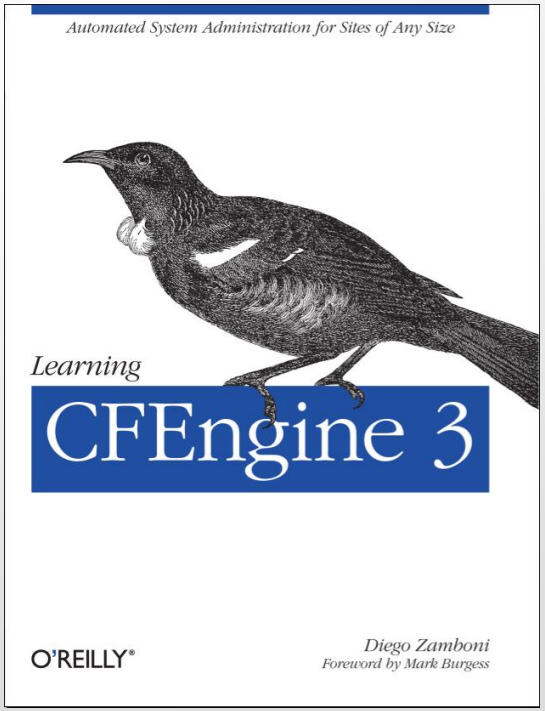
- V Community verzi neexistuje zpětná vazba (záměrně)
- Reporty o chybách jsou vytvářeny ručně
  - Používá se promise typu reports:
  - Každý klient odesílá e-mail administrátorovi
  - Použití více adres je možné jen změnou kódu politiky (konfigurace)
- V Enterprise verzi se zpětná vazba ukládá do centrální DB na serveru
- Řešení třetí strany
  - Delta Reporting od Evolve Thinking (od 4/2014)

Placená verze?

- Sběr zpětné vazby do centrální DB
- Mission Portal
- Reporting
- Windows nativně (instalace msi, registry, LDAP)
- Solaris
- Design Center GUI

Dokumentace

- Manuály, reference, příklady
  https://cfengine.com/docs/3.5/
- Někdy přehlednější
  https://cfengine.com/archive/manuals/
- Kniha
  http://shop.oreilly.com/product/0636920022022.do
- Diskusní skupina
  https://groups.google.com/forum/#!forum/help-cfengine
- Bugtracker
  https://cfengine.com/dev/projects/core



Závěr

- Promyšlený a logický koncept
- Jednotný deklarativní jazyk
  - Popisuje požadovaný stav, ne otrocky kroky jak ho dosáhnout
  - Široké vyjadřovací schopnosti, volnost
  - Naučit se psát politiky chce trochu cviku
- Minimalistická instalace
  - Nepotřebuje další podpůrný software
  - Funguje i na Raspberry Pi ☺
- V Community verzi chybí reporting (Enterprise?)
- CFEngine je velmi slibný nástroj do budoucna

Dotazy