

Gestione della configurazione su Linux. E non solo

L'amministrazione di sistema è l'insieme di tutte quelle attività di routine come installare e configurare programmi e driver, aggiornarli periodicamente, riconfigurarli quando cambiano i requisiti degli utenti, restaurare applicazioni e documenti dopo guasti o attacchi informatici. Sono compiti che di norma chi è solo «utente finale» di un computer potrebbe almeno in teoria ignorare, o addirittura (in azienda) avere il divieto esplicito di occuparsene per policy di sicurezza specifiche. Negli ultimi anni questa attività ha acquisito una nuova dimensione, o almeno una nuova variabile più o meno indipendente dalle altre, che la rende ancora più interessante e complessa allo stesso tempo. Ormai può capitare facilmente,

anche a chi non è responsabile di reti importanti in grandi aziende o enti pubblici, di essere l'amministratore di moltissimi computer. Accedere a macchine virtuali, anche centinaia, su Amazon Web Service (Aws) o altre piattaforme di *cloud computing* è sempre più facile, economico. C'è chi lo fa solo per motivi di studio e chi lo fa occasionalmente, per situazioni speciali come collaudare un nuovo programma, o analizzare una tantum enormi quantità di dati. Le cose non cambiano: sia perché l'informatica ha un ruolo sempre più critico, sia perché grazie al cloud computing aumenta a dismisura il numero di macchine da gestire: l'amministrazione di sistema è una pratica sempre più diffusa e critica. Aumentano il numero di persone coinvolte, il tempo

Come gestire decine di computer, aggiornare driver, configurazioni e installazioni con i programmi open source

necessario per i compiti di amministrazione e gli errori possibili.

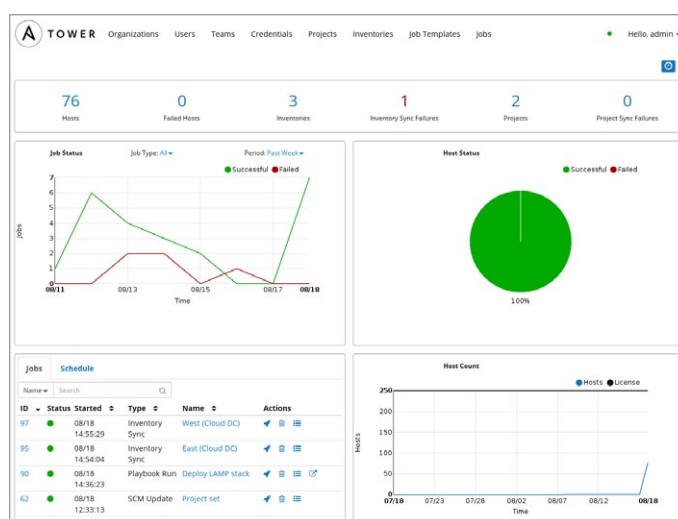
La gestione della configurazione (in Inglese Cm, cioè *Configuration Management*) è il livello superiore dell'amministrazione di sistema. "Fare Cm" significa gestire le ordinare attività di gestione informatica, automatizzandole e coordinandole il più possibile. Vediamo in queste pagine come gestire il Cm, su Linux e altri sistemi operativi, con strumenti Open Source. Non potremo ovviamente descrivere in dettaglio tutte le alternative possibili, vi invitiamo dopo questa introduzione a informarvi anche su prodotti di indubbio valore come Chef (www.getchef.com/chef) e Salt Stack (www.saltstack.com/community).

PRINCIPI DEI SISTEMI CM

Il mantra fondamentale del Cm è normalmente riassunto con qualche variante di questa battuta: sei un amministratore di sistema che, dopo un guasto, deve riconfigurare un server da zero con tutti i suoi programmi e parametri di sistema esattamente com'erano prima, fino all'ultimo dettaglio. Puoi farlo con **un solo comando**? Se la risposta è 'no' vuol dire che stai lavorando nel modo sbagliato. Perché tutto, ma proprio tutto, deve essere stato salvato da qualche *altra* parte, non sul server interessato. E deve esserlo non come semplice copia di file di configurazione, ma in un formato tale che sia possibile ricavarne automaticamente i comandi da impartire al server per tornare allo stato desiderato.

La prima cosa da sottolineare, in queste definizioni di Cm, è la differenza cruciale fra questa attività e i semplici backup di file da un lato, d'utente o di sistema, e i tanti meccanismi che automatizzano l'installazione *iniziale* di qualunque sistema operativo dall'altro. Prima di

L'interfaccia grafica di Ansible mostra in qualsiasi momento statistiche e diagrammi che riassumono lo stato di salute di tutti i computer collegati.



tutto, molto spesso non è sufficiente (o possibile) copiare fisicamente i file dove si trovavano in precedenza per garantire che un computer torni a comportarsi come prima.

In secondo luogo, anche dopo aver rimesso a posto tutti i file degli utenti e di sistema potrebbe essere necessario eseguire comandi direttamente alla console, con parametri magari semplici ma non costanti, quindi registrabili a priori in un file di configurazione.

Infine, anche se è proprio l'esempio da cui siamo partiti, il Cm non serve affatto solo *a posteriori*, cioè per recupero di computer già attivi, dopo guasti o attacchi. Il Cm serve per risparmiare tempo ed errori anche e soprattutto *a priori*, nell'installazione e configurazione iniziale di tanti nuovi computer. Anche se ciascun computer ha impostazioni leggermente diverse dagli altri, purché seguano qualche regola comune o almeno siano salvabili in database centrale. Il Cm è indispensabile, per gli stessi motivi, anche per tutte le operazioni di aggiornamento periodico.

In qualsiasi scenario di questo tipo (che, ripetiamo, potrebbe capitare temporaneamente anche a studenti o programmatori principianti, su piattaforme di *cloud computing*) sarebbe un suicidio organizzativo eseguire le procedure manualmente, cioè collegandosi a tutti i sistemi da gestire, uno alla volta.

Un sistema Cm adeguato aiuta e, allo stesso tempo, costringe, a effettuare tutte le modifiche a qualsiasi computer della propria rete in un unico luogo, in un unico modo. Successivamente il Cm garantisce che i comandi verranno automaticamente replicati sui computer sottostanti. Senza Cm, invece, si ha un'altissima probabilità di incorrere in almeno due tipi di problemi. Prima di tutto non si può avere la certezza di conoscere con sufficiente dettaglio la configurazione di tutti i propri computer. Anche registrando ogni singola azione per replicarla manualmente in caso di bisogno, si rischierebbe di non ritrovarsi con la stessa identica configurazione precedente. Non sarebbe possibile avere facilmente risposte affidabili e rapide a domande come "su quanti dei nostri computer sono installate versioni di sistema operativo vulnerabili agli ultimi attacchi informatici? In quali filiali si trovano?"

L'altro limite dell'amministrazione di sistema senza Cm è che diventa quasi impossibile essere sicuri di quali programmi stanno effettivamente operando su tutti i computer della propria rete, e di quanto siano aggiornati.

L'uso del Cm, infine, facilita notevolmente i collaudi di nuove configurazioni. Se si dispone di un server temporaneamente non utilizzato, lo si può sfruttare per collaudare il nuovo software in tutta tranquillità, sapendo che basterà un solo comando per rimettere tutto a posto come prima.

ANSIBLE

Anche limitandosi ai prodotti Open Source, ce ne sono diversi che permettono di svolgere attività di Configuration Management professionale, anche su grandi reti o su piattaforme di *Cloud Computing*. Quello che vogliamo presentare per primo è Ansible (www.ansible.com), perché ci sembra un ottimo compromesso fra ricchezza di funzioni e semplicità, sia in fase d'installazione sia durante l'uso. Buona parte del merito va al fatto che Ansible non ha un'architettura *client-server*, cioè basata su interazioni fra due componenti software ben distinti ma complementari, entrambi fatti apposta per svolgere un compito preciso. Ansible, invece, è *agent-less*: non c'è alcun software specifico ("agente", appunto) da

installare su ciascun computer da amministrare. Da quel lato, oltre ovviamente alla password di amministratore, nel caso di terminali Linux l'unico requisito è la presenza di un server Ssh per connessioni cifrate. Si tratta peraltro di un requisito davvero minimo, poiché questo software è

disponibile come pacchetto binario in tutte le distribuzioni Linux in circolazione e non poche lo installano per default. Le chiavi di cifratura necessarie per comunicare via Ssh senza password si possono caricare automaticamente sui terminali durante l'installazione. In caso contrario, le password Ssh si possono passare direttamente al programma Ansible sul computer di controllo, al momento della prima connessione.

In ogni caso, l'assenza di software specifico da installare sui terminali, magari remoti, significa meno lavoro, meno rischi e meno probabilità di guasti. Un discorso simile vale anche per controllare Windows con Ansible. In generale,

CM SÌ, MA SOLO SOTTO CONTROLLO!



C'è un lato, o meglio un corollario, del Configuration

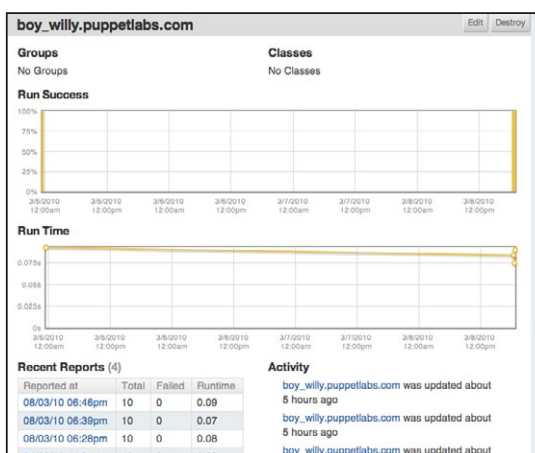
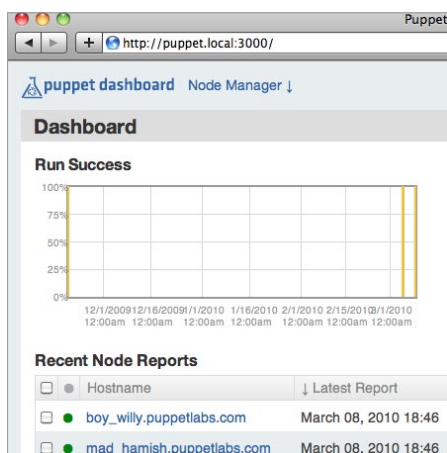
Management di cui occorre rendersi conto prima possibile, per evitare brutte sorprese e fatiche superflue. Forse è banale dirlo, ma... chi controlla i controllori? È inutile poter gestire molti computer registrando tutte le azioni effettuate su uno solo di loro, se quel computer speciale, o meglio tutti i suoi file rilevanti, non sono anch'essi gestiti a dovere.

Questo equivale a dire che la configurazione del software scelto per il Cm, e tutte le descrizioni dei computer che amministra, vanno inserite in un sistema di controllo del codice sorgente (come quelli descritti in questa rubrica, nel numero 278 di PC Professionale). Oltre ad avere la certezza di poter sempre recuperare qualsiasi versione di ciascun file in caso di cancellazioni accidentali, il controllo dei sorgenti offre altri due vantaggi. Uno è che permette di lavorare sempre allo stesso modo anche quando da computer diversi, in momenti diversi: basta replicare solo l'archivio di codice sotto controllo versioni su ciascun computer. Inoltre diventa anche facile combinare versioni diverse di vari file Cm, se necessario, senza rischiare di confondersi. Basta creare una vista apposita, proprio per quelle versioni, nel sistema di controllo.

Cloud computing

Ansible aiuta anche nella configurazione di server remoti e macchine virtuali

Questo dettaglio del pannello di controllo di Puppet, accessibile via Web, mostra i successi ottenuti nel tempo, cioè l'efficacia delle operazioni di Cm.



Quest'altro dettaglio di Puppet fornisce informazioni sullo stato di salute dei computer collegati e sugli ultimi controlli effettuati su ognuno di loro.

Ansible è stato progettato per essere sicuro ma minimale e con la minor curva di apprendimento possibile. Le istruzioni di configurazione sono scritte in un linguaggio più simile possibile a quello naturale, quindi facilmente aggiornabile e comprensibile anche per i principianti.

Questa semplicità d'uso è ovviamente relativa: per adoperare correttamente Ansible (o qualsiasi altro software Cm) bisogna conoscere le basi dell'amministrazione di sistema. Ma è comunque notevole, soprattutto considerando che non va a scapito della flessibilità. I comandi e le strutture dati di Ansible sono scalabili da una manciata di server in ufficio a centinaia di computer locali, remoti, virtuali e non, accessibili in parallelo per ottimizzare i tempi. Quando serve, l'utente può attivare tramite Ansible sistemi di bilanciamento del carico fra più server, oppure abilitare notifiche d'allarme diverse per ognuno di loro. Non è nemmeno necessario lanciare

sempre Ansible dallo stesso computer, né utilizzarne più installazioni indipendenti. Nel primo caso basta replicare tutti i dati e file necessari con qualche sistema di controllo delle revisioni (vedi il box a pagina 159) ovunque possano servire, mentre nel secondo è sufficiente lanciare Ansible con un file di configurazione generale specifico per la rete che si deve amministrare in quel momento. Il risultato finale sarà sempre lo stesso: l'invio a ogni macchina, tramite connessioni cifrate Ssh, dei comandi a basso livello da eseguire per portarsi nello stato desiderato. Svolgere tutto questo è possibile anche sfruttando lo sforzo collaborativo di sviluppatori e altri utenti Ansible: esiste una libreria ufficiale di quasi duecento moduli opzionali, ognuno pronto per eseguire le istruzioni relative a una specifica funzione, dall'autenticazione con vari protocolli alla configurazione di database. I moduli standard di Ansible supportano anche la tecnologia Open Source Docker e i servizi Amazon Web Service (Aws) per il cloud computing on demand. Con Docker (www.docker.com) si possono creare contenitori software, chiamati appunto Docker, per qualsiasi programma applicativo. I docker sono

“stagni”: una volta preparati per un certo sistema operativo, li si può copiare su qualsiasi computer e lanciare l'applicazione che contengono, senza alcuna installazione o configurazione. I servizi Aws sono invece molto più flessibili, perché danno accesso completo, tramite vari protocolli, a computer virtuali nei data center di Amazon.

Con i moduli giusti Ansible è già in grado di utilizzare quei protocolli, permettendo ai suoi utenti di non fare alcuna distinzione fra computer veri, locali, e quelli virtuali remoti. Anche le variazioni nel numero dei computer virtuali da controllare sono rilevate automaticamente. Se uno di essi dovesse diventare irraggiungibile per problemi di connessione, o se Amazon dovesse assegnarvi automaticamente nuovi server per far fronte a picchi di carico, la vostra installazione di Ansible se ne accorgerebbe automaticamente, intervenendo per garantire anche su quei computer la stessa configurazione di quelli già in attività. Nonostante questa loro versatilità, non sono i moduli l'arma più potente di Ansible. Questo è un titolo che spetta ai suoi *play* (rappresentazioni o cast), *playbook* (copioni) e ruoli. I play di Ansible definiscono gruppi di computer e i ruoli

Puppet per ogni OS

Il programma di gestione opera su Linux ma anche con Windows e OS X

The screenshot shows a web form for creating a new application cloud. It includes fields for Name, Description, Owned By (User or Team), Team, Type, Access Key, and Secret Key. There are 'Save' and 'Reset' buttons at the bottom.

Oltre al tempo, i sistemi Cm fanno guadagnare anche in sicurezza: Ansible, per esempio, può assegnare a ogni operazione un elenco di utenti autorizzati a svolgerla

RISORSE

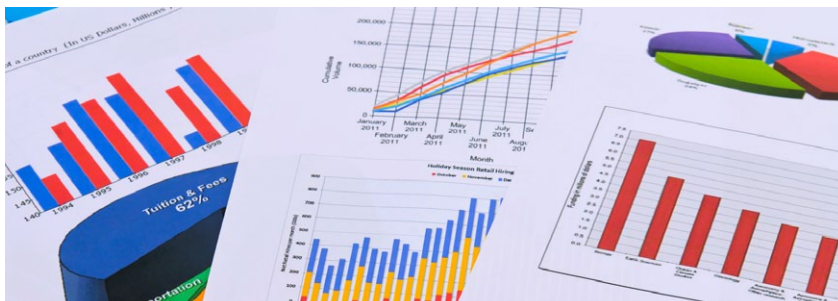
L'edizione inglese di Wikipedia include una pagina, abbastanza aggiornata, che confronta funzione per funzione le applicazioni Cm più popolari (http://en.wikipedia.org/wiki/Comparison_of_open-source_configuration_management_software). Ansible ha, in generale, un'eccellente documentazione: per orientarsi consigliamo di partire dalle sue Risposte Più Frequenti (<http://docs.ansible.com/faq.html>). Lo stesso vale per Puppet, che ha una sezione molto ricca del suo sito riservata alla documentazione (<http://docs.puppetlabs.com>): l'unica accortezza è non confondersi fra guide e funzioni valide solo nella versione commerciale del prodotto, Puppet Enterprise.

assegnati a ognuno di loro. Parlare di ruoli, cioè definire caratteristiche e obiettivi, permette all'utente di non occuparsi direttamente di quali comandi eseguire. A quelli ci pensa Ansible. I playbook sono script, sequenze di comandi che Ansible non potrebbe eseguire da solo perché richiedono password o altri parametri che solo l'utente può fornire. I playbook, che descrivono uno dei possibili stati di ogni computer, vengono raggruppati ed eseguiti in base ai ruoli assegnati.

E PER WINDOWS?

La flessibilità e semplicità di Ansible sono in arrivo anche sulle versioni più recenti di Windows. L'approccio di Ansible a questo problema è abbastanza diverso da quello usato per Linux, ma solo a basso livello. Non essendoci su questo sistema operativo né le stesse shell di Linux e Unix, né, di solito, un server capace di ricevere connessioni Ssh, Ansible utilizza script (e moduli) nel linguaggio PowerShell. Questo port è dichiarato ancora in stato *alfa*, ma è già utile per parecchie attività comuni: gestione di account utente e password, lancio di generici installer o script PowerShell remoti, e raccolta di statistiche sul funzionamento del sistema. Quel che è importante è che l'approccio ad alto livello a Windows è identico a quello per Linux: uso di ruoli e playbook, e nessun software dedicato da installare sui computer da mettere sotto Cm. L'altro pezzo da novanta, nel panorama moderno dei sistemi Cm Open Source, è Puppet (<http://puppetlabs.com>), che oltre a Linux, Unix e Windows supporta anche OS X. Puppet è un po' più complesso di Ansible, ma è simile nei concetti di base. Innanzitutto, anche Puppet permette di parlare quasi esclusivamente di utenti, gruppi di computer e loro funzioni, piuttosto che di file di configurazione o altri dati a basso livello. Come Ansible, anche Puppet dispone di un linguaggio descrittivo, che permette di spiegare nei termini più semplici possibile *dove* si vuole portare un certo computer o gruppo di computer, anziché quali comandi andrebbero eseguiti per ottenere lo stesso risultato.

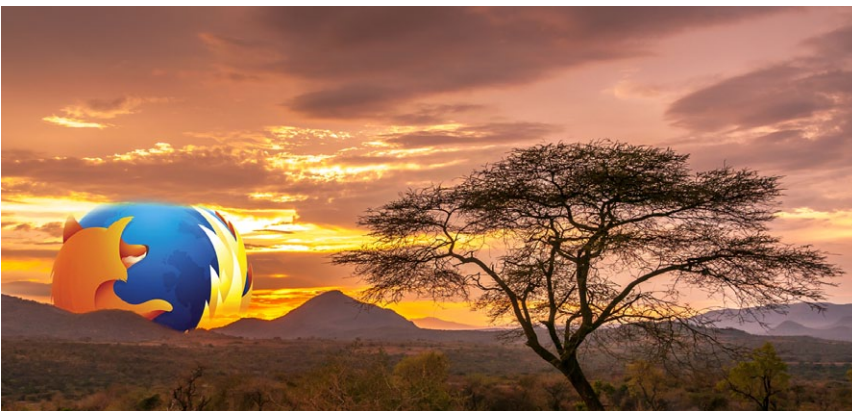
Un vantaggio di Puppet rispetto al concorrente è il numero di moduli già disponibili, oltre di dieci volte maggiore al momento, rispetto ad Ansible. I componenti automatizzano di tutto, dall'installazione e configurazione di server o firewall all'aggiornamento del software da archivi online.



LIBREOFFICE: NUOVE VERSIONI E NUOVE CERTIFICAZIONI

LibreOffice 4.3.3 e 4.37 continuano lo schema di rilasci della suite Open Source da ufficio ormai più popolare: sono entrambi versioni stabili, ma la prima introduce nuove funzioni, l'altra è di consolidamento, rivolta a chi alle novità preferisce stabilità, ritmi lenti e assenza di bachi. Ancora più importante è il lancio del nuovo programma di certificazione professionale, su due fronti: assistenza per la migrazione a LibreOffice e formazione generale sullo stesso pacchetto.

Chi riceverà la certificazione sarà accreditato come esperto di tutti gli aspetti della migrazione, dagli studi di fattibilità all'effettiva gestione dei progetti. I certificati abilitano anche a fornire ufficialmente supporto di cosiddetto "livello 3", quello che include lo sviluppo di nuove funzioni, o la risoluzione urgente di bachi su richiesta del cliente. Si tratta della prima iniziativa di questo tipo per un programma interamente sviluppato da una comunità di privati e aziende. Per ora solo chi è già membro della Document Foundation, l'ente no-profit che coordina lo sviluppo, potrà addestrarsi per ricevere la certificazione. Da aprile 2015 l'intero programma verrà aperto a tutti. Per ulteriori dettagli, consultare la pagina www.documentfoundation.org/certification.



MOZILLA SBARCA IN AFRICA CON FIREFOX OS

Il sistema operativo per smartphone chiamato Firefox OS, sviluppato dalla stessa fondazione che cura l'omonimo browser, dovrebbe presto diventare molto più comune fra gli utenti di diversi paesi africani. Mozilla Foundation ha infatti firmato accordi con tre diversi operatori mobili del continente, per fornire ai loro utenti terminali già equipaggiati, e permettendo quindi anche ai meno esperti di provare questa piattaforma. Contestualmente a questi accordi, la fondazione ha annunciato che sta valutando progetti di alfabetizzazione digitale, basati sugli stessi terminali con Firefox OS, con Brasile, India, Bangladesh e Kenya come primi destinatari.