

ПОРІВНЯННЯ ІНСТРУМЕНТІВ УПРАВЛІННЯ КОНФІГУРАЦІЄЮ ANSIBLE ТА CHEF

Маслій В.П.

Науковий керівник – доцент Власова В.О.

Харківський національний університет радіоелектроніки
(61166, Харків, просп. Науки, 14 каф. Інформаційно-мережевої інженерії,
тел. (096)595-65-40)

Tools for managing the Chef and Ansible buzz conceived to simplify the setup and servicing of the ten, hundreds and thousands of servers. You can automatically unload these configurations to your server in the middle of the server. This thesis will compare these configuration management tools.

Швидкий розвиток віртуалізації укупі зі збільшенням потужності серверів, які відповідають промисловим стандартам, а також доступність “хмарних обчислень” привели до значного зростання числа серверів які потребують управління. Зараз ми повинні управляти набагато більшою кількістю серверів, які можуть бути розподілені по всій земній кулі. У цей момент інструменти управління конфігураціями і вступають в гру. У багатьох випадках, ми управляємо групами однакових серверів, на яких запущені однакові програмами і сервіси. [1]

Ansible та Chef - це два найпопулярніших інструменти управління конфігурацією, які використовуються сьогодні. [2] Хоча обидва інструменти можуть виконувати багато одних і тих самих завдань, кожен з них має свої переваги, і вони виконують завдання помітно по-різному. Порівняймо ці інструменти за такими параметрам:

1. Архітектура

Chef працює по моделі клієнт-сервер, де є сервер Chef, в якому зберігаються ваші Cookbooks і на якому зареєстровані всі сервери або вузли, за які він відповідає. Цей Chef-сервер часто називають «концентратором». Самі вузли запускають клієнтське програмне забезпечення Chef, яке фактично виконує управління автоматизацією і конфігурацією в цій системі після отримання інструкцій від сервера Chef. Chef клієнти регулярно опитують сервер Chef на наявність змін в Cookbooks і, якщо зміни виявляються, виконують необхідні зміни стану в своїх системах. Основною перевагою цієї моделі є те, що вона вимагає дуже мало роботи для сервера Chef, оскільки все, що йому потрібно зробити, це надати клієнтам Chef відповідну інформацію при опитуванні. На відміну від Chef, Ansible не спирається на модель клієнт-сервер. Ansible не використовує агентів і нічого не вимагає для цільових хост-систем, крім ssh і Python. Це значно полегшує початок роботи з Ansible і звичайно, це означає, що на одну систему доводиться менше послуг, які ви повинні відслідковувати і обслуговувати.

Підсумок: почати роботу з Ansible легше, так як він не потребує агентів.

2. Мова

Іншою важливою відмінністю є Cookbooks написані на мові програмування Ruby в Chef і Playbooks написані на мові розмітки YAML в

Ansible. Для написання і читання Cookbooks необхідний досвід роботи з Ruby в той час як Playbooks написані на YAML, легкою для читання людиною мовою серіалізації, такий досвід не обов'язковий. Це значно спрощує створення нових Playbooks, а також дозволяє читати і розуміти Playbooks, написані кимось іншим, але має обмеження, що полягає в тому, що ви не завжди можете виконувати складні завдання з налаштування таким чином, щоб це було природно. Chef дозволяє простіше вирішувати завдання зі складною логікою роботи. Насправді, складні завдання в Ansible іноді можуть здатися дуже «хакерськими», і вам іноді доведеться покладатися на Jinja2 для цих цілей. З огляду на, що Ansible почав розробку тільки в 2012 році, можна очікувати, що з часом випадковий «хакерський» обхідний шлях поступиться місцем більш природнім рішенням в більш пізніх випусках програмного забезпечення.

Підсумок: Playbooks в порівнянні з Cookbooks набагато простіше створювати і розуміти, але вони обмежені в складності завдань конфігурації, які ви можете виконувати з їх допомогою.

3. Статична і динамічна інвентаризація

Ще однією перевагою Ansible перед Chef є те, як він керує своєю системою хост-систем. З Chef хости повинні бути індивідуально зареєстровані на сервері Chef. Однак Ansible пропонує кілька варіантів для джерела інвентаризації: простий INI-файл з хостами, класифікованими по розділах; база даних; або інше джерело, можна опитувати, для визначення того, які на ньому сучасні системи. Така гнучкість дозволяє Ansible використовувати запаси, які є статичними або динамічними.

Підсумок: на відміну від Chef, Ansible може використовувати динамічні і статичні списки хост-систем.

Проте, якщо ви повинні вибрати між ними, спочатку ретельно продумайте свої власні потреби і порівняйте їх з тим, що пропонує кожне рішення.

Переваги Ansible. Швидкий та простий при розгортанні та написанні Playbooks. CLI підтримує майже всі мови програмування. Гарна безпека яка забезпечується використанням SSH ключів.

Недоліки. Відносно новий та недостатньо тестований. Недопрацьований GUI та погана підтримка Mac Os.

Переваги Chef. Більш розвинений та має гарну документацію. Має багато модулів для вирішення складних задач доволі просто.

Недоліки. Досвід роботи з Ruby. Використання формату JSON замість більш дружнього для користувача YAML.

Список джерел

1. Огляд: Puppet, Chef, Ansible, Salt [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/211306/>

2. Chef vs Ansible [Електронний ресурс] – Режим доступу до ресурсу: <https://www.loomsystems.com/blog/single-post/2017/07/20/chef-vs-ansible-comparison-of-configuration-management-tools>