

Chef vs Puppet vs Ansible vs Saltstack"

Table of Contents

1. Disponibilité	1
2. Facilité d'installation	2
3. La gestion	2
4. Évolutivité	3
5. Langue de configuration	3
6. Interopérabilité	3
7. Coût	4

Métrique	Chef	Puppet	Ansible	Saltstack
Disponibilité	□	□	□	□
Facilité d'installation	Pas très facile	Pas très facile	facile	Pas très facile
Gestion	Pas très facile	Pas très facile	facile	Pas très facile
Evolutivité	Très évolutif	Très évolutif	Très évolutif	Très évolutif
Langage de configuration	DSL (Ruby)	DSL (Ruby)	YAML (Python)	YAML (Python)
Interopérabilité	Haute	Haute	Haute	Haute
Coût	13700 \$	11200 \$ - 19900 \$	10 000 \$	15 000 \$ (app)

1. Disponibilité

Laissez-moi comparer chef vs Puppet vs ansible vs saltstack sur la base de la disponibilité. Tous les outils sont hautement disponibles, ce qui signifie qu'il y a plusieurs serveurs ou plusieurs instances présentes. Dites, si votre maître ou serveur principal tombe en panne, il y a toujours un serveur de sauvegarde ou le maître différent pour prendre sa place. Jetons un coup d'œil à chaque outil un par un:

Chef - Lorsqu'il y a une défaillance sur le serveur principal, c'est-à-dire le chef serveur, il a un serveur de sauvegarde pour remplacer le serveur principal.

Puppet - Il a une architecture multi-maître , si le maître actif tombe en panne, l'autre maître prend le lieu maître actif.

Ansible - Il s'exécute avec un seul noeud actif, appelé l'instance principale. Si le primaire descend, il y a une instance secondaire pour prendre sa place.

Saltstack - Il peut avoir plusieurs maîtres configurés. Si un maître est arrêté, les agents se connectent avec l'autre maître de la liste. Par conséquent, il a plusieurs maîtres pour configurer les séides de Saltstack.

2. Facilité d'installation

Quand je parle de facilité d'installation, permettez-moi d'ajouter mon expérience personnelle parce que quand j'installais le chef, la Puppet et le saltstack, j'ai rencontré quelques problèmes mais quand j'installais Ansible, c'était juste comme une promenade de gâteau. Alors concentrons-nous sur chaque outil un par un:

Chef - Chef a une architecture maître-agent. Le serveur Chef s'exécute sur l'ordinateur maître et le client Chef s'exécute en tant qu'agent sur chaque ordinateur client. En outre, il existe un composant supplémentaire appelé station de travail, qui contient toutes les configurations testées et ensuite transmises au serveur du chef central. Par conséquent, ce n'est pas si facile.

Puppet - Puppet a aussi une architecture maître-agent. Le serveur de Puppets s'exécute sur la machine maître et les clients de Puppets s'exécutent en tant qu'agent sur chaque machine client. Après cela, il existe également une signature de certificat entre l'agent et le maître. Par conséquent, ce n'est pas aussi facile.

Ansible - Seul le maître s'exécute sur la machine du serveur, mais aucun agent ne s'exécute sur la machine cliente. Il utilise la connexion ssh pour se connecter aux systèmes clients ou aux nœuds que vous souhaitez configurer. La machine client VM ne nécessite aucune configuration particulière, d'où la rapidité d'installation!

Saltstack - Ici, Server est appelé en tant que maître du Saltstack et les clients sont appelés en tant que serviteurs de Saltstack qui s'exécutent en tant qu'agents dans la machine cliente.

3. La gestion

Avant d'expliquer la différence entre ces outils sur la base de la gestion, laissez-moi vous dire Puppet et chef suit les configurations de traction et Ansible et Saltstack suit la configuration de poussée. Vous devez vous demander quelles sont ces configurations? En configuration push, toutes les configurations présentes dans le serveur central seront poussées vers les nœuds alors que, dans la configuration pull, les nœuds esclaves tireront automatiquement toutes les configurations du serveur central sans aucune commande.

1. Chef - Vous devez être un programmeur pour gérer les configurations car il offre des configurations en Ruby DSL. Le client extrait les configurations du serveur.
2. Puppet - Pas très facile à gérer les configurations car il utilise son propre langage appelé Puppet DSL (Domain Specific Language). Le client extrait les configurations du serveur. Il est très orienté système-administrateur et il existe une exécution à distance non-immédiate.
3. Ansible - Facile à apprendre pour gérer les configurations car il utilise YAML, c'est-à-dire Yet Another Markup Language qui ressemble beaucoup à l'anglais. Le serveur pousse les configurations à tous les nœuds. Bon pour une application en temps réel et une exécution à distance immédiate.

4. Saltstack - Facile à apprendre pour gérer les configurations car il utilise aussi YAML. Le serveur pousse les configurations à tous les clients. Exécution à distance immédiate

4. Évolutivité

Tous les quatre outils sont hautement évolutifs. Supposons que vous ayez besoin de configurer environ 50 nœuds aujourd'hui, et 500 demain. Pas de problème avec ces outils. Il peut gérer une grande infrastructure, il vous suffit de spécifier l'adresse IP et le nom d'hôte des nœuds que vous souhaitez configurer et le reste de la tâche sera géré par ces outils. Par conséquent, tous ces outils sont hautement évolutifs.

5. Langue de configuration

Chef Chef utilise Ruby Domain Specific Language (Ruby DSL). Il a une courbe d'apprentissage abrupte et son développeur orienté.

Puppet - Puppet utilise son propre langage de domaine de Puppets (Puppet DSL). Ce n'est pas très facile à apprendre et son système d'administration orienté.

Ansible - Ansible utilise YAML c'est-à-dire Yet Another Markup Language (Python). Il est assez facile à apprendre et orienté vers l'administrateur. Python est intégré dans la plupart des déploiements Unix et Linux de nos jours, de sorte que la configuration de l'outil est plus rapide.

Saltstack - Saltstack utilise également YAML (Python). C'est encore facile à apprendre et orienté administrateur.

Ensuite, allons de l'avant et comparons chef vs Puppet vs ansible vs saltstack sur la base de l'interopérabilité.

6. Interopérabilité

Dans ces outils, maître ou serveur principal ou vous pouvez également dire machine de contrôle, doit être sur Linux / Unix mais leurs esclaves ou les nœuds qu'ils doivent configurer peuvent être sur Windows. Jetons un coup d'oeil à chaque outil un par un:

Chef - Chef Serveur fonctionne uniquement sous Linux / Unix, mais Chef Client et Workstation peuvent également être sous Windows.

Puppet - Puppet Master fonctionne uniquement sous Linux / Unix mais Puppet Agent fonctionne également sous Windows.

Ansible - Ansible supporte aussi les machines Windows mais le serveur Ansible doit être sur une machine Linux / Unix.

Saltstack - Salt Master fonctionne uniquement sous Linux / Unix mais les serveurs de Salt peuvent aussi travailler sur Windows.

7. Coût

Les coûts d'entreprise pour les outils de configuration sont les suivants:

Chef - Chef Automate vous donne tout ce dont vous avez besoin pour construire, déployer dans \$ 137 nœud / annuel.

Puppet - Les prix pour les Puppets vont de 112 \$ par nœud / année avec un plan de soutien standard à 199 \$ par nœud / année avec le plan de prime.

Ansible - Le prix de la tour Ansible pour les opérations informatiques standard jusqu'à 100 nœuds est de 10 000 \$ / an. Cela inclut le support 8 * 5 tandis que les offres premium 24 * 7 supportent 14000 \$ / an.

Saltstack - Le coût pour Saltstack Enterprise par 100 nœuds est de 15,00 \$ / an (environ). Vous pouvez contacter le support pour le prix de l'abonnement annuel actuel.