

	название системы	поколение	год выпуска	сильные стороны	слабые стороны	push / pull стратегия	агент	Язык конфигураций	Тестирование	Комьюнити
1	CFEngine	I	1993	Система даёт гарантии, что она правильно настроена, и проверяет стабильному состоянию системы, а не изменениям.	1. все изменения и пожелания весьма жёстко контролировались Марком Бюргесом 2. Большие возможности сделали систему относительно сложной в изучении, язык выглядел запутанным, на его изучение придётся потратить время. 3. не смотря на наличие многостраничных руководств, в которых расписаны многие тонкости работы с CFEngine - написаны они разработчиками, которые часто «забывают» указать отдельные «мелочи», всплывающие во время настройки, или вовремя не вносят изменения.	Pull	CFEngine Agent	язык C		наличие многостраничных руководств на английском языке, в которых расписаны многие тонкости работы с CFEngine. Но, как водится, написаны они самими разработчиками, которые часто «забывают» указать отдельные «мелочи», всплывающие во время настройки, или вовремя не вносят изменения.
2	Chef	II	2009	1. используется для упрощения настройки и обслуживания серверов компании. 2. интегрируется с облачными платформами, такими как Internap, Amazon EC2, Google Cloud Platform, Oracle Cloud, OpenStack, IBM Cloud, Microsoft Azure и Rackspace для автоматического выделения ресурсов и настройки новых машин. 3. Chef написан на Ruby и имеет Ruby DSL. Используется вся мощь Ruby. 4. В поставку входят инструменты для тестирования Cookstyle, ChefSpec, Chef InSpec и Test Kitchen для проверки, что код Chef Infra выполняет что было планировано перед развёртыванием на серверах Прода. 5. Огромное комьюнити и самый большой набор инструментов среди всех SCM. 6. Использование ресурсов, представленных в cookbooks сообществах, или написание собственных, подходящих для собственной инфраструктуры.	1. использование агента на управляемом хосте 2. Ruby является полноценным языком программирования, что подразумевает порог вхождения 3. по причине огромной документации - она местами неясная и может быть неточной 4. недостает функциональных команд "push"	Pull	Chef-клиент	Chef написан на Ruby и имеет Ruby DSL		У Chef огромное комьюнити и самый большой набор инструментов среди всех SCM. https://community.chef.io/
3	Puppet	II	2005	1. Благодаря ограничениям встроенного языка Puppet описание конфигурации хорошо читается и наглядно не только для разработчиков, но и системных администраторов. 2. при необходимости можно использовать Ruby. 3. Для написания манифестов для Puppet есть IDE на базе Eclipse — Gepetto, предусмотрены режимы для пользователей Vim и Emacs. 4. Описание конфигурации значительно упрощается благодаря большому числу модулей, позволяющих разворачивать распространенные приложения (например, MySQL), а также настраивать саму систему, например сетевые интерфейсы, временную зону и т. п.	1. Многие модули перед использованием требуют доработки и/или исправления ошибок. Например, может не проверяться текущая операционная система и устанавливаться пакет ‘apache2’ в CentOS (правильное название пакета в этом случае — ‘httpd’), как будто это Ubuntu 2. использование агента на управляемом хосте 3. наличие большого числа модулей затрудняет выбор 4. встроенный язык может вызывать проблемы, связанные с зависимостями	Pull	puppet agent	Конфигурация описывается на специальном декларативном предметно-ориентированном Ruby-подобном языке		У приложения Puppet самый большой опыт. Неудивительно, что его разработчик смог организовать самое большое собственное сообщество и набрать солидную базу пользователей

4	SaltStack	III	2011	<ul style="list-style-type: none"> 1. имеет как безагентную архитектуру, которая работает в push-режиме при помощи Salt SSH, так и клиент-серверную архитектуру 2. один из самых гибких SCM на данный момент 3. использует автоматизацию в реальном времени 4. использует YAML-подобный язык 5. отлично масштабируется 	<ul style="list-style-type: none"> 1. нельзя назвать готовым к использованию продуктом. Является скорее Фреймворком. 2. использование агента на управляемом хосте 3. web-интерфейс является самым слабым среди рассматриваемых SCM 4. не хватает инструментов для детальных отчетов 	Pull	<ul style="list-style-type: none"> 1. может иметь безагентную архитектуру в push-режиме при помощи Salt SSH 2. клиент-серверную архитектуру, где есть Salt Master и агент Salt Minion 	YAML-подобный язык, код очень похож на Ansible		Salt уже успела обрасти сторонними модулями и проектами, а также сильным community
5	Ansible	III	2012	<ul style="list-style-type: none"> 1. самый быстрый 2. простой в использовании 3. может использоваться без установленных агентов на управляемых хостах 4. имеет YAML-подобный DSL 5. модули могут быть написаны почти на любом языке 	<ul style="list-style-type: none"> 1. отсутствует поддержка клиентов под Windows 2. плохо масштабируется 3. Код модулей выполняется при импорте, поэтому сложно протестировать код в модулях 	push	не требуется	YAML-подобный DSL	Код модулей выполняется при импорте, поэтому сложно протестировать код в модулях	Достаточно большое комьюнити, которое поддерживает Модули (сейчас их уже больше шести тысяч), а также активно участвует в развитии продукта