

# Автоматизация мастера в CloudForms 4.1

Peter McGowan

Version v1.0, 2016-06

# Table of Contents

Предисловие .....	1
A Brief Word on Terminology .....	1
Who Should Read This Book .....	1
Navigating This Book .....	2
Online Resources .....	7
Conventions Used in This Book .....	7
Работа с автоматическим .....	9
обеспечение виртуальных машин .....	10
Настройка виртуальных машин .....	11
Работа с услугами .....	12
Выход на пенсию .....	13
Интеграция .....	14

# Предисловие

Red Hat Cloudforms и его «восходящие» сестры управляют и сестрами являются мощными платформами облака, которые позволяют нам эффективно управлять нашей виртуальной инфраструктурой и инфраструктурой в качестве облаков услуги (IAAS). Значительная часть этой эффективности заключается в автоматизации многих повседневных задач, которые в противном случае потребовали бы ручного участия, или трудоемких и, возможно, склонных к ошибкам повторяющихся шагов.

Эта книга представляет собой введение и руководство по работе с функцией *Automate* CloudForms 4.2 и соответствующим выпуском *ManageIQ euwe*.

Автоматизация упрощает нашу жизнь и повышает нашу эксплуатационную эффективность. Это позволяет нам делать такие вещи, как:

- Устранение многих ручных решений и операций, связанных с обеспечением виртуальных машин и облачных экземпляров.
- Баланс нагрузки Наши виртуальные машины по всей нашей виртуальной инфраструктуре, чтобы соответствовать способу работы нашей организации, будь то логично (например, центр затрат, отдел), операционная (например, среда жизненного цикла инфраструктуры) или категориальная (например, роль сервера или характеристика виртуальной машины).
- Создайте каталоги услуг, чтобы позволить нашим пользователям предоставлять виртуальные машины с одной кнопкой *order*.
- Создайте автоматические облачные приложения, где новые виртуальные машины динамически предоставляются по требованию.
- Управляйте нашим полным жизненным циклом виртуальной машины.
- Интегрируйте наш рабочий процесс обеспечения виртуальной машины с более широким предприятием, например, автоматически регистрируя новые виртуальные машины со стороны спутникового сервера Red Hat.
- Реализуйте интеллектуальные рабочие процессы пенсионных процессов виртуальной машины, которые отменяют ресурсы, такие как IP-адреса, и нерегистрация от служб каталогов.

## A Brief Word on Terminology

Эта книга относится к *automate* как к возможностям CloudForms или ManageIQ или функции продукта, а также *Automation* как то, что позволяет нам автоматизировать. *Automation Engine* позволяет нам создавать интеллектуальные рабочие процессы автоматизации и запустить *automation scripts*, написанные в Ruby или Ansible Playbooks.

## Who Should Read This Book

Эта книга понравится администраторам облачных или виртуализации, которые заинтересованы в автоматизации частей их виртуальной инфраструктуры или облачных

вычислений. Несмотря на то, что в основном нацелены на тех, кто знаком с CloudForms или Manageiq, многие из концепций и терминов, таких как *orchestration* и *automation Workflows*, будут легко поняты даже тем, кто не знаком с продуктом.

Автомат может быть одним из наиболее сложных аспектов инструментов для освоения. Практикующий требует необычного сочетания навыков; Знакомство с традиционными «инфраструктурными» концепциями, такими как виртуальные машины, гипервизоры и сети арендаторов, а также стиль для сценариев в Ruby и мастерство модели объекта программирования. Однако нет черной магии, и все навыки могут быть изучены, если нам покажут путь.

Книга предполагает разумный уровень компетентности с рубиновым языком со стороны читателя. Существует много хороших онлайн-учебных пособий Ruby, в том числе Codecademy's <http://www.codecademy.com/tracks/ruby>® для программы в Ruby].

Книга также предполагает удобный уровень опыта работы и знакомства функций веб-пользовательского интерфейса (WebUI), которые либо CloudForms, либо Manageiq, в частности, Insight, Control, Tagging и Provision VMS через **LifeCycle** → **Provision VMS** Точка записи. Многие из этих функций будут автоматизированы, когда мы следуем примерам в книге, и поэтому полезно понимание того, почему тегирование полезно (например).

#### NOTE

Как CloudForms, так и Manageiq являются веб-приложениями, поэтому взаимодействие преимущественно через WebUI на основе браузера. Мы используем терминал командной строки только тогда, когда первоначально настраиваем устройство CloudForms, или при устранении неполадок или проверки журналов.

## Navigating This Book

Книга разделена на семь частей.

### Part I "Working With Automate"

Глава 1: *introduction to CloudForms и Manageiq*, устанавливает сцену и описывает возможности CloudForms и Manageiq как платформы управления облаком.

Глава 2, *introduction в автоматизация данных DataStore*, выводит нас в экскурсию по объектам, с которыми мы работаем, когда используем возможности автоматизации CloudForms и Manageiq.

Глава 3, *Writing и запуск наших собственных автоматических сценариев*, знакомит нас с написанием сценариев автоматизации в Ruby, с простым «Привет, мир!» пример.

Глава 4, *using схемы переменных*, показывает, как мы можем использовать схему нашего экземпляра для хранения и получения переменных.

Глава 5, *working с виртуальными машинами*, демонстрирует, как работать с объектом виртуальной машины автоматизации и как запустить сценарий автоматизации с пользовательской кнопки в интерфейсе веб-пользователя.

Глава 6, *peeping в Hood*, представляет некоторую теорию фона о Rails *Models* и о том, как CloudForms и Manageiq абстрагируют их как *service models*, с которыми мы работаем, когда пишем наши сценарии автоматизации.

Глава 7, *\_ \$ evm и The Workspace\_*, выводит нас в экскурсию по полезным методам `$ evm`, которые мы часто используем при сценарии, такие как `$ evm.vmdb`` и `$ evm.object`.

Глава 8, *а Практический пример: обеспечение соблюдения правил анти-аффинности*, является реальным примером полноцветного фильма о том, как мы могли бы использовать методы, изученные до сих пор для внедрения правил анти-аффинности в нашей виртуальной инфраструктуре, основанных на тегах.

Глава 9, *using Tags из Automate*, подробно описывает, как мы можем создавать, назначать, читать и работать с тегами из наших сценариев Automation.

Глава 10, *investigative Debugging*, обсуждает, как мы можем выяснить, какие объекты доступны нам при сценарии. Это полезно как с точки зрения расследования при разработке сценариев, но и для отладки наших сценариев, когда что-то не работает, как и ожидалось.

Глава 11, *ways ofing automate*, показывает нам различные точки ввода рабочего процесса в автоматизированное хранилище данных. Это также иллюстрирует, как мы можем программно определять, как был вызван нашим скриптом автоматизации, что позволяет нам создавать повторные сценарии.

Глава 12, *Requests and Tasks*, иллюстрирует, как больше расширенных операций автоматизации разделены на этап *request*, который требует административного утверждения для прогресса на этап *task*. Соответствующие объекты запроса и задачи описаны, и их использование сравнивается.

Глава 13, *state Machines*, знакомит нас с государственными машинами, и как мы можем использовать их для разумной последовательности наших рабочих процессов.

Глава 14, *\_ МОРНАЯ ПРЕДОСТАВЛЕННАЯ СХЕМА и ЧЕЛОВЕКИ ЭНТЕРСОВОГО ЧЕЛОВЕКА\_*, обсуждается более продвинутое, но реже используемое схемы и функции экземпляра; Сообщения, утверждения, коллекции и экземпляр `.missing`.

Глава 15, *EVENT Processing*, описывает, как CloudForms и Manageiq реагируют на внешние события, такие как выключение виртуальной машины, и прослеживает последовательность обработки событий через автоматическое хранилище данных. Он также показывает, как Automate управляет собственными внутренними событиями, такими как `request_started`.

## Part II "Provisioning Virtual Machines"

Глава 16, *provision Airtual Machine*, представляет концепцию обеспечения виртуальной машины, самой сложной операции автоматизации, выполняемой CloudForms и Manageiq.

Глава 17, *\_ The Provisioning Profile\_*, описывает, как ссылается профиль подготовки для определения начальной групповой обработки, которая выполняется в начале операции обеспечения виртуальной машины.

Глава 18, *PAPPORVAL*, показывает, как работает рабочий процесс одобрения и как мы можем регулировать критерии автоматического одобрения, такие как количество виртуальных машин, которые подлежат предоставлению, или объем хранения, в соответствии с нашими потребностями.

Глава 19, *quota Management*, рассказывает подробности механизма обработки квот CloudForms и Manageiq, и о том, как он позволяет нам устанавливать квоты для арендаторов или групп.

Глава 20, *\_ Опции HASH\_*, объясняет важность структуры данных, называемой *options hash*, и как мы можем использовать его для извлечения и хранения переменных для настройки операции по обеспечению виртуальной машины.

Глава 21, *\_ The Provisioning State Machine\_*, обсуждает этапы на государственной машине, которая регулирует последовательность операций, связанных с обеспечением виртуальной машины.

Глава 22, *Customising Virtual Machine Provisioning*, является практическим примером, показывающим, как мы можем настроить статусную машину и включить наши собственные методы, чтобы добавить второй жесткий диск во время операции обеспечения виртуальной машины.

Глава 23, *virtual machine именованное во время Provisioning*, объясняет, как мы можем настроить логику *naming*, которая определяет имя, данное вновь предоставленной виртуальной машине.

Глава 24, *Virtual Machine Machine во время Provisioning*, объясняет, как мы можем настроить логику *placement*, которая определяет местоположение хоста, кластера и данных для данных для нашей недавно предоставленной виртуальной машины.

Глава 25, *\_ The Provisioning Dialog\_*, описывает диалоги Webui, которые приводят к приглашению на параметры, которые требуются до того, как может быть предоставлена новая виртуальная машина. В главе также объясняется, как диалоги могут быть настроены для расширения дополнительных диапазонов для элементов, таких как размер памяти, или для представления диалога с ограниченным на заказ определенным группам пользователей.

Глава 26, *Virtual Machine Provisioning Objects*, подробно описывает четыре основных объекта, с которыми мы работаем, когда пишем Ruby Scripts, чтобы взаимодействовать с процессом обеспечения виртуальной машины.

Глава 27, *creating requestions Provisioning Programmaly*, показывает, как мы можем инициировать операцию по обеспечению виртуальной машины из сценария автоматизации вместо интерфейса веб-пользователя.

## **Part III "Automation using Ansible Tower"**

Глава 28, *Automation с использованием ANSIBLE*, вводит некоторые Ansible Concepts и описывает функции башни, которые мы используем при автоматизации с использованием Ansible, таких как запасы, роли и шаблоны работы.

Глава 29, *toverse Automate Components*, описывает автоматические компоненты данных и модели обслуживания, которые используются при использовании Ansible для задач автоматизации.

Глава 30, *Running Ansible Tower Work с Button*, является практическим примером создания шаблона задания на сервере Ansible Tower, а затем запуск его на виртуальной машине с кнопки WebUI в CloudForms. Глава 31, *integrating со спутником 6 во время Provisioning*, является примером, показывающим, как можно использовать Ansible Playbook для регистрации недавно созданной виртуальной машины с помощью Red Hat Satellite 6, либо в качестве *host* или *content host* (или оба).

## Part IV "Working with Services"

Глава 32, *service Dialogs*, представляет компоненты, которые составляют диалог `_service`, включая элементы, которые могут быть динамически заполнены методами Ruby.

Глава 33, `_ Служба`, обеспечивающая государство `Machine_`, обсуждает этапы на государственной машине, которая регулирует последовательность операций, связанных с созданием услуги.

Глава 34, *catalog {item, bundle} инициализация*, описывает два конкретных экземпляра машины состояния услуг, которые были разработаны для упрощения процесса создания каталога услуг *items* и *bundles*.

Глава 35, `_ одобрение и квота_`, показывает рабочий процесс утверждения для услуг и то, как новый механизм обработки квот также применяется к услугам.

Глава 36, *creating a Cataloge Cataloge Atem*, является практическим примером, показывающим, как создать элемент каталога услуг для предоставления виртуальной машины.

Глава 37, *creating a Service Catalog Bundle*, является практическим примером, показывающим, как создать пакет каталога услуг из трех виртуальных машин.

Глава 38, *Assible Tower Services*, описывает компоненты автоматизации данных и модели сервисов, которые используются при создании сервисов, которые запускают шаблоны рабочих мест Ansible Tower.

Глава 39, *creating a Ansible Tower Service Cataloge u Bundle*, иллюстрирует, как создать пакет каталога обслуживания, который обеспечивает новую виртуальную машину, и впоследствии запускает его шаблон работы башни для настройки компонентов программного обеспечения лампы.

Глава 40, *service Objects*, представляет собой разоблачение различных объектов, которые работают за кулисами, когда предоставляется элемент каталога услуг.

Глава 41, *log-анализ во время предоставления услуг*, представляет собой пошаговую прогулку, отслеживая строки, записанные на `_automation.log` на разных этапах операции по предоставлению услуг. Это может помочь наше понимание нескольких уровней происходящей одновременной активности машин состояния.

Глава 42, *service иерархии*, иллюстрирует, как услуги могут содержать другие услуги, и мы можем организовать наши сервисные группы в иерархии для организационного и удобства управления.

Глава 43, *service Reconfiguration*, описывает, как мы можем создавать реконфигурируемые услуги. Они способны принимать параметры конфигурации во время заказа через диалоговое окно службы, а впоследствии могут быть перенастроены с помощью новых параметров конфигурации через тот же диалог службы.

Глава 44, *service Tips and Chriecks*, упоминает некоторые полезные советы, которые следует помнить при разработке услуг.

## Part V "Retirement"

Глава 45, *\_Virtual Machine* и экземплярный пенсионер, обсуждает процесс выхода на пенсию для виртуальных машин и экземпляров.

Глава 46, *service пенсионер*, обсуждает процесс выхода на пенсию для услуг.

## Part VI "Integration"

Глава 47, *calling Automate из Restful API*, показывает, как мы можем сделать внешние вызовы *into* CloudForms или Manageiq для запуска экземпляров автоматизации через API Restful. Мы также можем вернуть результаты нашему абоненту таким образом, позволяя нам создать наши собственные конечные точки Pseudo-API на двух платформах.

Глава 48, *automation запрос Appraval*, объясняет, как настроить поведение по умолчанию по умолчанию для запросов на автоматизацию, чтобы неадминистраторы могли отправлять запросы API Restful без необходимости административного утверждения.

Глава 49, *calling External Services*, показывают различные способы позвонить *out* из автоматизации для интеграции с нашим более широким предприятием. Это включает в себя изготовление исходящих вызовов для отдыха и мыла, подключение к базам данных MySQL и взаимодействие с OpenStack с использованием GEM FOG.

## Part VII "Miscellaneous"

Глава 50, *tency and Automate*, описывает модель аренды CloudForms/Manageiq и то, как нам иногда нужно адаптировать наши сценарии автоматизации для работы с концепцией аренды.

Глава 51, *distributed Automation обработка*, описывает, как Automate был разработан, чтобы быть горизонтально масштабируемым. В главе описывается механизм, с помощью которого запросы на автоматизацию распределяются между несколькими приборами в регионе.

Глава 52, *Argument Передача и обработка*, объясняет, как аргументы передаются и обрабатываются внутренне методами автоматизации для каждого из различных способов, которые мы называли им до этого момента в книге.

Глава 53, *miscelly tips*, закрывает книгу с помощью некоторых полезных советов по



разработке автоматизации метода.

## Online Resources

Существует несколько онлайн -ресурсов, о которых должен знать любой студент CloudForms или Manageiq Automate.

### Official Documentation

Официальная документация для CloudForms здесь: <https://access.redhat.com/documentation/en/red-hat-cloudforms/>

Официальная документация по управлению здесь: <http://manageiq.org/documentation/>

### Code Repositories

Одним из лучших источников справочного материала является превосходная коллекция кодов *cloudforms\_essentials*, поддерживаемая Квином Морей из Red Hat ([https://github.com/ramrexx/cloudforms\\_essentials](https://github.com/ramrexx/cloudforms_essentials)). Это содержит множество полезных образцов кода, и многие из примеров в этой книге произошли из этого источника.

Существует также очень полезный консалтинг Red Hat (<https://github.com/rhtconsulting>) github, поддерживаемый несколькими консультантами Red Hat.

### Fora

Проект Manageiq проводит форум *manageiq talk* по адресу <http://talk.manageiq.org>

### Blogs

Есть несколько блогов, которые имеют хорошие облачные формы и статьи, связанные с управлением, включая некоторые полезные *notes из Field*. К ним относятся:

- CloudForms сейчас (<http://cloudformsblog.redhat.com/>)
- Блог Кристиана (<http://www.jung-christian.de>)
- Laurent Domb OSS Blog (<http://blog.domb.net/>)
- Все открывается (<http://allthingsopen.com/>)
- Tigeriq (<http://www.tigeriq.co/>)

## Conventions Used in This Book

В этой книге используются следующие типографские соглашения:

### *Italic*

Указывает новые термины, URL -адреса, адреса электронной почты, имена файлов, имена пути и имена объектов в автоматическом хранилище данных, значения поля схемы

## Bold

Указывает компоненты WebUI, имена событий, имена поля схемы

## Постоянная ширина

используется для списков программ, а также в параграфах для ссылки на такие элементы программы, как имена переменной или функций, базы данных, типы данных, переменные среды, операторы и ключевые слова.

## Постоянная ширина Bold

Показывает команды или другой текст, который должен быть напечатан буквально пользователем.

## ***\_ Постоянная ширина Italic \_***

Показывает текст, который следует заменить на поставленные пользователем значения или значениями, определяемыми контекстом.

<b>NOTE</b>	Этот значок означает общую ноту.
-------------	----------------------------------

<b>TIP</b>	Эта икона означает наконечник или предложение
------------	---

<b>WARNING</b>	Этот значок указывает предупреждение или осторожность.
----------------	--

# Работа с автоматическим

Часть I представляет объекты и концепции, с которыми мы работаем, когда разрабатываем наши сценарии автоматизации. Некоторые из глав содержат практические примеры кодирования, чтобы помочь процессу обучения, в то время как другие более теоретические и содержат справочную информацию, чтобы помочь построить общее понимание объектной модели. Unresolved directive in <stdin> - include::chapter1.asciidoc[] Unresolved directive in <stdin> - include::chapter2.asciidoc[] Unresolved directive in <stdin> - include::chapter3.asciidoc[] Unresolved directive in <stdin> - include::chapter4.asciidoc[] Unresolved directive in <stdin> - include::chapter5.asciidoc[] Unresolved directive in <stdin> - include::chapter6.asciidoc[] Unresolved directive in <stdin> - include::chapter7.asciidoc[] Unresolved directive in <stdin> - include::chapter8.asciidoc[] Unresolved directive in <stdin> - include::chapter9.asciidoc[] Unresolved directive in <stdin> - include::chapter10.asciidoc[] Unresolved directive in <stdin> - include::chapter11.asciidoc[] Unresolved directive in <stdin> - include::chapter12.asciidoc[] Unresolved directive in <stdin> - include::chapter13.asciidoc[] Unresolved directive in <stdin> - include::chapter14.asciidoc[] Unresolved directive in <stdin> - include::chapter15.asciidoc[]

# обеспечение виртуальных машин

Обеспечение виртуальной машины, вероятно, является самой сложной работой, которая выполняется с помощью механизма автоматизации.

Часть II представляет шаги и рабочие процессы, связанные с обеспечением виртуальных машин, и как мы можем настроить их для наших собственных целей. Он использует все функции автоматизации, которые мы обсуждали до сих пор в книге. Unresolved directive in <stdin> - include::chapter16.asciidoc[] Unresolved directive in <stdin> - include::chapter17.asciidoc[] Unresolved directive in <stdin> - include::chapter18.asciidoc[] Unresolved directive in <stdin> - include::chapter19.asciidoc[] Unresolved directive in <stdin> - include::chapter20.asciidoc[] Unresolved directive in <stdin> - include::chapter21.asciidoc[] Unresolved directive in <stdin> - include::chapter22.asciidoc[] Unresolved directive in <stdin> - include::chapter23.asciidoc[] Unresolved directive in <stdin> - include::chapter24.asciidoc[] Unresolved directive in <stdin> - include::chapter25.asciidoc[] Unresolved directive in <stdin> - include::chapter26.asciidoc[] Unresolved directive in <stdin> - include::chapter27.asciidoc[] Unresolved directive in <stdin> - include::chapter28.asciidoc[]

# Настройка виртуальных машин

В части II мы увидели, как мы можем предоставить виртуальные машины от Webui. CloudForms 4.1 и Manageiq *DARGA* добавили возможность запускать задания конфигурации через Ansible Tower. Это позволяет нам обеспечить и настраивать виртуальные машины от автоматизации.

В части III мы узнаем о настройке виртуальных машин с использованием Ansible и Ansible Tower. Unresolved directive in <stdin> - include::chapter29.asciidoc[] Unresolved directive in <stdin> - include::chapter30.asciidoc[] Unresolved directive in <stdin> - include::chapter31.asciidoc[] Unresolved directive in <stdin> - include::chapter32.asciidoc[] Unresolved directive in <stdin> - include::chapter33.asciidoc[] Unresolved directive in <stdin> - include::chapter34.asciidoc[] Unresolved directive in <stdin> - include::chapter35.asciidoc[] Unresolved directive in <stdin> - include::chapter36.asciidoc[] Unresolved directive in <stdin> - include::chapter37.asciidoc[] Unresolved directive in <stdin> - include::chapter38.asciidoc[] Unresolved directive in <stdin> - include::chapter39.asciidoc[]

# Работа с услугами

В части II мы увидели, как мы можем предоставить виртуальные машины от Webui. Этот процесс включает в себя введение значений для многих параметров диалога подготовки, таких как наше имя и адрес электронной почты, а также желаемая конфигурация виртуальной машины; Например, количество процессоров, объем памяти, формата диска и VLAN. В части III мы также видели, как мы можем настроить виртуальные машины, используя Ansible Tower из Automate.

CloudForms и Manageiq также позволяют нам создавать каталоги услуг, которые позволяют нам предоставлять и настраивать виртуальные машины с одной кнопки «Порядок» \* (см. << P3I1 >>).

1. Speemle Service Каталог image::images/pt3\_ss1.png["Simple Service Catalog"]

В части IV мы узнаем о создании и использовании услуг. Unresolved directive in <stdin> - include::chapter40.asciidoc[] Unresolved directive in <stdin> - include::chapter41.asciidoc[]

# Выход на пенсию

Процесс выхода на пенсию для виртуальных машин, экземпляров и сервисов был существенно переписан для CloudForms 3.1 (Manageiq *botvinnik*). Часть V описывает текущие пенсионные рабочие процессы для виртуальных машин и услуг. Unresolved directive in <stdin> - include::chapter42.asciidoc[] Unresolved directive in <stdin> - include::chapter43.asciidoc[] Unresolved directive in <stdin> - include::chapter44.asciidoc[]

# Интеграция

Одной из мощных особенностей CloudForms и Manageiq является *integration* или его способность организовать и координировать и координировать с внешними службами как часть рабочего процесса. Эти услуги могут включать в себя решение по управлению корпоративным IP -адресом (IPAM), например, систему билетов или базу данных управления конфигурацией (CMDB).

В части VI рассматриваются возможности интеграции автоматизации. Unresolved directive in <stdin> - include::chapter45.asciidoc[] Unresolved directive in <stdin> - include::chapter46.asciidoc[] Unresolved directive in <stdin> - include::chapter47.asciidoc[]