Интеграция марионеточных ролей и профилей

Содержание

Введение

Cisco Intelligent Automation for Cloud 4.1 теперь включает управление приложениями. С Приложением, Настраивающим поддержку, можно управлять приложениями на сервер или виртуальную машину (VM). Cisco IAC 4.1 позволяет Начальную загрузку Настраивать (т.е. инициализация приложения для действительного и физических серверов.) Или агент Марионетки или Повара автоматически загружен на настраиваемый VM.

Перед началом работы

Требования

Прежде, чем делать попытку этой конфигурации, гарантируйте соответствие этим требованиям.

Удостоверьтесь, что Cisco IAC 4.1 или позже установлен, настроен, и работающий в поддерживаемых версиях (или посмотрите Матрицу совместимости Cisco Intelligent Automation for Cloud для подробных данных) перед началом.

Марионеточное Предприятие должно быть установлено на устойчивых, выделенных серверах, которые могут обработать количество агентов, которым они должны будут служить. Консольная роль может быть установлена на том же сервере как марионеточное ведущее устройство или может быть разделена. Поскольку мы используем развертывания агента/ведущего устройства, необходимо подготовить сеть к трафику Марионетки. Мы ожидаем, что межсетевой экран должным образом настроен, ваш марионеточный главный

сервер должен позволить входящие соединения на порту, который вы выбрали; по умолчанию это 8140, и узлы агента должны быть в состоянии соединиться с ведущим устройством на том порту.

Используемые компоненты

Сведения в этом документе основываются устойчивый, выделили Марионеточные Корпоративные серверы, чтобы обработать умеренный веб - трафик и выполнить с высокой загрузкой процессора фоновые задачи. Для получения информации о компонентах, см. Марионеточную страницу требований Предприятия.

Условные обозначения

Дополнительные сведения об условных обозначениях в документах см. Cisco Technical Tips Conventions.

Установка

1. Загрузите и проверьте целевое Марионеточное предприятие (PE) tarball.
2. Распакуйте tarball. (Выполните tar-xvf <tarball>.)
3. Из каталога установщика PE выполните sudo./puppet-enterprise-installer.
4. Когда предложено, выберите “Yes” для установки установочных пакетов. На этом этапе установщик PE запустит Web-сервер и перейдет к вебу - адресу: https://<устанавливают имя хоста платформы>:3000. Гарантируйте, что хост достижим с портом 3000. Примечание: Оставьте свое подключение терминала открытым, пока не будет завершена установка; в противном случае установка откажет.
5. Скопируйте адрес в свой браузер.
6. Когда предложено, примите запрос безопасности в своем браузере. Затем вы должны быть взяты к начальной странице установщика.
7. На начальной странице нажмите, Let's начинают работу.
8. Затем, вас попросят выбрать ваш тип развертывания. Выберите Monolithic.
9. Предоставьте следующую информацию о марионеточном главном сервере:Марионеточный основной FQDN: предоставляет полное доменное имя сервера, на котором вы устанавливаете PE; например, ведущее устройство. пример. com.Псевдонимы DNS: предоставляет список разделенных запятой значений узлов агента псевдонимов, может использовать для достижения ведущему устройству; например, 'ведущее устройство'.Имя пользователя SSH: вводит имя пользователя SSH для пользователя, соединяющегося с марионеточным ведущим устройством; в этом случае, 'root'.
10. Когда предложено о поддержке БД, выберите Установку параметра по умолчанию PostgreSQL для меня.
11. Предоставьте следующую информацию об администраторе консоли PE:Консольный адрес электронной почты суперпользователя: предоставьте адрес, который вы будете использовать для регистрации к консоли как администратор.Консольный пароль суперпользователя: создайте пароль для регистрационного имени консоли; как обозначено, пароль должен составить по крайней мере восемь символов.
12. Нажмите кнопку Submit (Отправить).
13. На подтвердить странице плана рассмотрите информацию вы, если, и, если это

выглядит корректным, нажимают Continue.

14. На странице проверки установщик проверит различные элементы конфигурации (например, корректны ли учетные данные SSH, существует достаточно дискового пространства, и если ОС является тем же для различных компонентов). Если нет никаких нерешенных вопросов, нажмите Deploy теперь.
15. Установщик тогда установит и настроит Марионеточное Предприятие. Это, возможно, также должно установить дополнительные пакеты от репозитория вашего ОС. Этот процесс может занять до 10-15 минут. Когда установка будет завершена, сценарий установщика, который работал в терминале, закроет себя.

!--- конфигурацию

Марионеточная основная конфигурация

После установки Марионеточного Ведущего устройства с IAC необходимо проверить следующие аспекты конфигурации сервера:

1. Удостоверьтесь, что вы включаете “hiera_config” опцию в основной блок марионеточного файла конфигурации (пример файла конфигурации может быть найден ниже).
2. Удостоверьтесь, что ваши модули расположены в папке, заданной в “basemodulepath” параметре puppet.conf файла.
3. В hiera файле конфигурации включайте “: datadir:” parameter. (пример файла конфигурации может быть найден ниже).

```
Example of working Pupet configuration file:

[main]
 certname = pupm.server.local
    dns_alt_names = pupm.server.local,pupm
    vardir = /var/opt/lib/pe-puppet
    logdir = /var/log/pe-puppet
    rundir = /var/run/pe-puppet
    basemodulepath = /etc/puppetlabs/puppet/modules:/opt/puppet/share/puppet/modules
    server = iac-qe-pupm.tidalsoft.local
    user  = pe-puppet
    group = pe-puppet
    archive_files = true
    archive_file_server = pupm.server.local
    hiera_config = /etc/puppetlabs/puppet/hiera.yaml

[master]
    certname = pupm.server.local
    ca_name = 'Puppet CA generated on pupm.server.local at 2014-07-22 22:39:18 -0500'
    reports = console,puppetdb
    node_terminus = console
    ssl_client_header = SSL_CLIENT_S_DN
    ssl_client_verify_header = SSL_CLIENT_VERIFY
    storeconfigs = true
    storeconfigs_backend = puppetdb

[agent]
    report = true
```

```
    classfile = $vardir/classes.txt
    localconfig = $vardir/localconfig
    graph = true
    pluginsync = true
    environment = production

Example of working hiera configuration file:

    ---
    :backends:
      - yaml

    :yaml:
      :datadir: /etc/puppetlabs/puppet/hieradata

    :hierarchy:
      - "nodes/%{fqdn}"
      - common
```

Роли и профили

Марионетка основана на принципалах абстракции: поставщики абстрагированы типами, ресурсы абстрагированы классами, классы абстрагированы моделями, и модули абстрагированы профилями. Высший уровень абстракции является ролями.

Роли являются просто наборами профилей, которые предоставляют разумное сопоставление между человеческой логикой и логикой технологии. Таким образом, “правила”, окружающие дизайн Ролей, могут быть упрощены как:

1. Роль включает один или несколько профилей для определения типа сервера
2. Профиль включает и управляет модулями для определения логического технического стека
3. Модули управляют ресурсами
4. Модули должны только быть ответственны за управление аспектами компонента, для которого они записаны

Получить профили и роли обнаружило в портале, они должны быть расположены в определенных модулях, названных “ролью” и “профилем”. Пример местоположений:

- Декларации ролей - $basemodulepath/role/manifests
- Декларации профиля - $basemodulepath/profiles/manifests
- $basemodulepath – путь местоположения модуля задан в марионеточном файле конфигурации.

```
Example of roles (each role should be located in individual manifest):
class role {
  include profile::base
}

class role::www inherits role {
  # All WWW servers get tomcat
  include profile::tomcat
}

class role::www::dev inherits role::www {
  include profile::webserver::dev
```

```
  include profile::database
}

class role::www::live inherits role::www {
  include profile::webserver::live
}

class role::mailserver inherits role {
  include profile::mailserver
}

Example of profiles (each profiles should be located in individual manifest):
class profile::base {
  include networking
  include users
}

class profile::tomcat {
  class { "jdk": }
  class { "tomcat": }
}

class profile::webserver {
  # Configuration for all webservers
  class { "httpd": }
  class { "php": }
  class { "memcache": }
}

class profile::webserver::dev inherits profile::webserver {
  Class["php"] {
    loglevel   => "debug"
  }
}

class profile::webserver::live inherits profile::webserver {
  # Any live webserver specific stuff here
}

class profile::database {
  class { "mysql": }
}

class profile::mailserver {
  class { "exim": }
}
```

IAC: соединение с марионеткой

1. Выберите Марионеточный тип элемента платформы.
2. Введите дружественное имя для соединения.
3. Предоставьте имя хоста для соединения или IP-адреса.
4. Введите описание для этого соединения.
5. Введите вход в систему SSH, который будет иметь разрешения для выполнения команд Puppet.
6. Введите пароль для входа в систему. Также предоставьте секретный ключ в разделе "Дополнительных параметров".
7. Введите пароль повторно.

8. Как альтернатива обеспечению пароля для соединения, можно скопировать содержание секретного ключа (pem) файл.Дополнительные параметры для начальной загрузки, прокси и файла закрытого ключа.
9. Задайте базовый URL для загрузки Марионеточного пакета установщика Предприятия. Расположение по умолчанию является репозиторием PuppetLabs.
10. Задайте альтернативный путь модуля:По умолчанию обнаружение использует modulepath, как определено в puppet.conf. Можно указать обнаружение к альтернативному пути, такому как GIT, работающий копия. В определении альтернативного пути можно использовать $environment для динамичной вставки среды в путь.
11. Путь классификации узлов Hiera:Новые узлы классифицированы с помощью hiera yaml файлы, и, по умолчанию, сохранены к Марионеточному Основному местоположению, заданному первой папкой, заданной в пути модуля. Можно было сохранить эти файлы в другом месте, возможно отделиться от модулей. В определении пути hiera можно использовать $environment для динамичной вставки среды в путь. Помните, что это местоположение должно совпасть с тем, что определено в вашем hiera.yaml файле в каталоге конфигурации Марионеточного Ведущего устройства.
12. Выберите информацию Bootstrap/Proxy для Операционной системы:Linux:Дополнительно, задайте адрес прокси-сервера для использования при настройке серверов Linux.Можно также задать список обходных адресов для прокси.Дополнительно, введите "известного" пользователя/пароль для начальной загрузки программного обеспечения конфигурации на новых узлах Linux; когда пароль *не* задан в заказе относительно root, применяется.
13. Windows:Дополнительно, задайте адрес прокси-сервера для использования при настройке Windows Server.Можно также задать список обходных адресов для прокси.Дополнительно, введите "известного" пользователя/пароль для начальной загрузки программного обеспечения конфигурации на узлах новых окон; если не заданный, будут использоваться учетные данные для присоединения к домену.

Примечание: Загружающиеся пароли отображены в открытом тексте и не безопасны. Это для начальной буквы, загружающейся только. Это должно быть изменено во время конфигурации. Этот пароль только используется, когда IAC не может настроить пароль гостевого пользователя, и пароль в исходном шаблоне должен использоваться.

Проверка

В настоящее время для этой конфигурации нет процедуры проверки.

Дополнительные сведения

- Марионеточные компоненты и требования
- Марионеточное Краткое руководство по началу работы Предприятия
- Марионеточная документация Предприятия
- Интеллектуальная автоматизация Cisco для облака