

文章

[Michael Lei](#) · 六月 23, 2021 阅读大约需 13 分钟

使用 Ansible 自动化部署 Caché 应用程序 - 第 1

部分 Ansible 帮助我解决了快速部署 Caché 和应用程序组件以进行数据平台基准测试的问题。您可以使用相同的工具和方法来建立您的测试实验室、培训系统、开发或其他环境。如果在客户站点部署应用程序，可以将大量部署自动化，并确保系统、Caché 和应用程序的配置符合您的应用程序最佳做法标准。

概述

作为一名技术架构师，我们的团队职责之一是在不同供应商的硬件和操作系统上对 InterSystems 数据平台进行基准测试。通常，基础架构是预发布版，在必须归还或移交给其他人之前，我们的时间有限，因此，快速准确地设置基准测试，让我们有尽可能多的时间来做真正的基准测试工作，这一点至关重要。

多年来，我们通过 shell 脚本小程序来自动执行许多基准测试安装任务，并从速查表和检查清单中剪切和粘贴，但此类操作非常密集，而且容易出错，特别是有许多服务器并且在不同的操作系统之间切换时 - 在 SLES 11、Red Hat 6、Red Hat 7、AIX 等操作系统上安装或使用服务的差异可能很微小，让人厌烦。

在研究了几个可用于自动化配置和管理系统的软件选项之后，我选择了 Ansible 来执行预置数据平台、应用程序和基准测试组件的任务。需要注意的是，我并没有规定 Ansible 是部署和配置的 THE 解决方案。在选择 Ansible 之前，我研究了其他工具（如 Puppet 和 Chef）的功能和操作。如果您的组织已经在使用其他工具，您可以使用它们 — 我在 Ansible 中使用的方法和命令等等应该可以转换到其他软件中，我希望这些帖子可以帮助您，不管您使用的是什么工具。

这是本系列的第一个帖子，将介绍在部署 InterSystems 数据平台应用程序时如何使用 Ansible。本帖介绍如何通过安装 Caché 打下基础，下一帖将扩展解决方案以包括应用程序安装，包括使用 %installer 类。本帖涵盖：

- Ansible 的概述和安装
- Ansible 便于管理和扩展的布局。
- 同时在一个或多个服务器上安装 Caché。

什么是 Ansible ?

通过 Ansible 可以在配置一个或多个服务器的同时将复杂的任务自动化，并可以非常简单地添加新服务器。任务会设计成幂等（您可以在同一台服务器上多次运行相同的脚本，得到的服务器配置将是相同的）。

我选择 Ansible 执行预置任务的一个关键原因是它对系统的要求最低（Python 2.7，Linux 服务器上自带），而且它是一个自包含解决方案 — Ansible 代码只安装在控制服务器上，并使用推送架构，通过 OpenSSH 在目标服务器上运行命令和脚本。所预置的服务器上不需要任何代理。作为对比，Chef 和 Puppet 采用拉取架构，软件在客户端服务器（Web、数据库等）上加载，并且客户端不断轮询主服务器以查找更新。Ansible 的推送架构也适合按照您的计划需求逐步实施服务器。

Ansible 是开源的，由社区维护。Ansible, Inc 从 2015 年开始为 Red Hat 所拥有。Ansible, Inc 有一个高级的生命周期产品 (Ansible Tower)，并提供收费的支持和培训，不过本帖中的所有内容均使用开源命令行版本。还有一个活跃的社区 (Ansible Galaxy)，您可以从中下载许多预制的解决方案来完成大量任务，如安装 Web 服务器、ftp、kerberos，不胜枚举。例如，对于完整的基准测试部署项目，我包括了一个下载的 Apache 模块，并自定义成在 RHEL、SLES 或 Solaris（以及其他平台）上安装和配置 Apache 2.x。

Ansible 的下载和安装说明可以在 Ansible 网站和 Github 上找到。
如果您有问题或希望做出贡献，可以访问活跃社区。

<https://www.ansible.com/get-started>
<http://docs.ansible.com>

安装 Ansible

本帖中的示例已经在运行 Red Hat 7.0 和 7.2 的虚拟机上进行了测试 - 我也在我的安装了 Centos 7 的笔记本电脑上使用 virtual box 和 vagrant 对 Ansible 控制器服务器进行了初始测试。 Caché 不需要安装在控制器上，所以您的操作系统选择要多于 Caché 支持的平台列表。
为了简单起见，我使用了当前适用于 Red Hat 的 rpm 版本的 Ansible (Ansible 1.9.4)，更高的版本可以从 GitHub 获取。

在示例中，我安装的是 cache-2015.2.2.805.0-Inxrhx64，但相同的常规过程也适用于 HealthShare 或 Ensemble 发行版。 您将在后面看到，我们使用特定文件名、目录路径等变量来参数化安装选项。

在第一个帖子中，我将任务削减为基本的 Caché 安装，因此大多数任务是独立于平台的。 当 Ansible playbook 启动时，首要任务之一是获取目标机器的清单 — 操作系统、接口卡、内存详细信息、CPU 数量、磁盘布局等，当运行命令以从目标上运行的实际命令（例如，Red Hat 上的 service start httpd 与 SLES 上的 /etc/init.d/apache2 restart）中提取 Ansible 脚本命令时，将使用目标操作系统的这些信息。

我假定您已读过说明，并且已按照您的平台说明在控制机上安装了 Ansible。

<http://docs.ansible.com/ansible/introinstallation.html>

Ansible 必须使用 Linux 系统作为控制器，但目标系统可以是 Linux 或 Windows。 有关 Windows 目标的更多信息，请参见 Ansible 文档。

<http://docs.ansible.com/ansible/introwindows.html>

控制器系统安装示例：在 RHEL/CentOS 7 64 位上安装 Ansible

在 Red Hat 或 CentOS 上，必须先安装 epel-release (Extra Packages for Enterprise Linux) RPM，其中包含 Ansible。 epel 项目面向主要 Linux 发行版设计，提供了许多有用的开源软件包（网络、系统管理、监视等）。

```
[root@localhost tmp]# wget http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
:
:
[root@localhost tmp]# rpm -ivh epel-release-7-5.noarch.rpm
:
:
[root@localhost tmp]# yum --enablerepo=epel info ansible
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Installed Packages
Name : ansible
Arch : noarch
Version : 1.9.4
Release : 1.el7
Size : 7.0 M
Repo : installed
From repo : epel
Summary : SSH-based configuration management, deployment, and task execution system
URL : http://ansible.com
License : GPLv3+
```

Description :

: Ansible is a radically simple model-driven configuration management,
: multi-node deployment, and remote task execution system. Ansible works
: over SSH and does not require any software or daemons to be installed
: on remote nodes. Extension modules can be written in any language and
: are transferred to managed machines automatically.

```
[root@localhost tmp]#  
[root@localhost tmp]# sudo yum install ansible  
:  
:  
[root@localhost tmp]# ansible --version  
ansible 1.9.4  
configured module search path = None
```

很好... 准备开始!

Ansible 方面

关于不同的 Ansible 组件（清单、Playbook、模块、角色等）的用法，应查看 Ansible 文档。

为了简化管理以及避免使用大而复杂的脚本文件，使用了预定义的目录结构和搜索路径。
在本帖中，我将使用一个采用 Ansible 建议的文件结构，当您考虑构建更大的安装示例时，可以将本结构用作模型。

所使用的 Ansible 模块带有注释和自我说明，可以在 Github 上获取。 下载文件并通读以了解工作流程。

<https://github.com/intersystems/ansible-deploy-cache>

我们的示例的基本目录包含以下文件；

- ansible.cfg：对 Ansible 默认值的更改。
- 清单：定义和描述工作环境。例如服务器名称/IP。
- <各种>.yml 文件：这些文件描述了将为特定服务器角色运行的任务集。

术语

为了使后面的讨论更清晰，下面是一些 Ansible 术语的快速解释。

模块是用于创建对系统执行的自动化操作的构建块。每个模块都为特定任务构建，可以使用参数更改该任务。
例如复制文件、创建用户、运行命令、启动服务等。目前，默认的 Ansible 安装已包含 400 多个模块，另外还有更多来自社区的模块，您也可以创建您自己的模块。

模块组合在一起生成 play 和 playbook，作为执行自动化工作流程的一种方式。一个 Play 可以有多个任务，一个 Playbook 可以有多个 Play。

角色允许您组合 Playbook。角色可以被视为根据目标服务器使用情况分组在一起的服务器组件配置。
在本帖的示例中，角色构建配置层以构建服务器。

在我的基准测试设置中，我有以下角色来构建服务器：

- hservercommon：配置操作系统，安装 Apache，安装 Caché。
- webserver：复制 Web 文件（csp、html、js 等），针对应用程序配置 Apache。
- generator：复制文件，创建和配置 webstress 生成器数据库、命名空间、全局映射等。
- dbserver：复制文件，配置数据库服务器系统设置、应用程序数据库、命名空间、全局映射等。

可以组合这些角色来构建不同的服务器类型：

- `hsservercommon + webserver + generator = webstress` 生成器服务器。
- `hsservercommon + webserver =` 应用程序 Web 服务器。
- `hsservercommon + dbsevrer =` 数据库服务器。

角色的组成以及每个角色中包含的配置将非常特定于要部署的应用程序。本帖中的示例将使用最小的任务集，并假定操作系统已预先配置，但是，使用 Ansible 和 Galaxy 上的模块可以实现更复杂和全功能的系统配置。

关于安装 Caché 的说明

我写了几个例子来介绍一些有趣和有用的特性，下面是一些精华部分。注意：这些示例可以用作 InterSystems 数据平台（Caché、HealthShare 和 Ensemble）的安装指南。我写的示例是安装 HealthShare，但 HealthShare 和 Caché 示例中的操作是相同的。

```
./testserver/roles/hsservercommon/tasks/main.yml
```

这是配置操作系统、安装 Apache、安装 Caché 等常见任务的主线脚本。本帖中对其进行了删减，使其只包含用于 Red Hat 的文件，以及只安装和配置 Caché。您可以看到，Ansible 在启动后已经在 `ansible*` 变量（包括 `ansibleosfamily`）中保存了操作系统信息，我们可以在脚本中使用这些变量做出决策。

```
./testserver/roles/hsservercommon/tasks/configure-healthshare2015.yml
```

这是用于安装 Caché 的主脚本。浏览该脚本，会看到针对目标执行的任务的逻辑工作流程，包括：

- 创建操作系统用户和组。
- 从控制器上的清单文件夹复制安装文件。
- 解压缩安装文件。
- 使用静默安装方式安装 Caché（请参见下面的注释）。
- 复制 Caché 密钥文件
- 设置默认 Caché 实例
- 重启 Apache
- 重启 Caché

Caché 的静默安装有几个选项，包括：

- 使用 `parameters.is` 文件。模板 `.isc` 文件由以前的安装创建，可以按原样使用，也可以修改。
- 使用 `cinstallsilent` 以及环境中设置的键值对。
- 使用 `%installer` 类。

在此示例中，我选择了使用 `installsilent`，但是我还包括了一个注释掉的使用参数文件的备用方法，以说明如何在 Ansible 中使用模板文件（请参见 `/roles/hsservercommon/templates/parametershs20152rh64.isc`）。

在以后的帖子中，我将说明如何使用 `%installer` 类安装 Caché 以及设置数据库和命名空间。有关安装选项的详细信息，请参见 Caché 在线文档，社区中也有一个非常好的帖子介绍了 `%installer` 类的使用。

当您想要安装并配置 Caché，以将 CSPGateway 与除了旧版本 Caché 内部的 Apache 版本以外的 Web 服务器一起使用时，参数文件很有用。自 Caché 2016.1 起，`%installer` 提供此功能。

```
./testserver/roles/hsservercommon/tasks/setupRedHat.yml
```

包含此示例是为了说明如何使用系统特定变量 (`ansible*`) 和设置操作系统变量。

```
**./testserver/roles/hsservercommon/vars/**
```

变量文件包含键:值对形式的变量，如您所见，这是一种在不同环境和情况下重复使用相同脚本的方法。

运行 Caché 安装

对于此示例，我假定系统可用，并按如下方式进行设置。

1. 控制器已安装 Ansible，并且以下目录填充了来自 Github 的文件和结构。
 - `*/testserver/***`：包含清单、.yml 文件等等的目录树。包括...
 - `./testserver/DistributionFiles/Cache`：（包含 Caché 分发包和 `cache.key` 的清单）。
2. 目标机器已安装 Red Hat 和 Apache。

您需要编辑以下文件来为您的测试环境自定义安装。

1. `inventorytest`

您需要编辑测试服务器名称或 IP 地址。

2. `./testserver/roles/hsservercommon/vars/healthshare2015.yml`

您必须编辑路径以适合您的测试环境。查看目标服务器的以下路径：

- `commoninstallbasepath`：清单文件将复制到该位置后解压，并运行 Caché 安装。
- `ISCPACKAGEINSTALLDIR`：Caché 安装目录

如果目标服务器上不存在这些目录路径，将会创建。

注：自动化部署的特性之一是并行构建多个服务器。如果清单文件中有多个服务器，在各个目标服务器上将并行运行每个步骤，直至该步骤完成，然后再在组中的每个服务器上开始下一个步骤。

如果任何服务器上有步骤失败，脚本将停止。您将看到一条错误消息，帮助您更正问题。

更正错误后，只需从头重新运行 — 这是脚本的一个关键特性 — 脚本设计成幂等。幂等的意思是，在某个步骤运行某个模块（例如复制文件）时，如果文件已经存在，则该步骤不会重新运行，脚本只是继续下一个步骤。copy 之类的模块有参数可以设置为强制复制，但这不是默认设置。

仔细检查脚本会发现，在某些情况下使用了“creates”参数，例如：

```
- name: unattended install of hs using cinstall_silent
shell: >
  ISC_PACKAGE_INSTANCENAME="{{ ISC_PACKAGE_INSTANCENAME }}"
  ISC_PACKAGE_INSTALLDIR="{{ ISC_PACKAGE_INSTALLDIR }}"
  ISC_PACKAGE_UNICODENAME="{{ ISC_PACKAGE_UNICODENAME }}"
  ISC_PACKAGE_INITIAL_SECURITY="{{ ISC_PACKAGE_INITIAL_SECURITY }}"
  ISC_PACKAGE_MGRUSER="{{ ISC_PACKAGE_MGRUSER }}"
  ISC_PACKAGE_MGRGROUP="{{ ISC_PACKAGE_MGRGROUP }}"
  ISC_PACKAGE_USER_PASSWORD="{{ ISC_PACKAGE_USER_PASSWORD }}"
  ISC_PACKAGE_CACHEUSER="{{ ISC_PACKAGE_CACHEUSER }}"
  ISC_PACKAGE_CACHEGROUP="{{ ISC_PACKAGE_CACHEGROUP }}" ./cinstall_silent
  chdir="{{ common_install_base_path }}/{{ hs_install_unpack_path }}"
  args:
  creates: "{{ ISC_PACKAGE_INSTALLDIR }}/cinstall.log"
```

上面一节使用 creates 参数告诉 Ansible 模块（在本例中是 shell 模块），此操作创建 `cinstall.log` 文件。

如果模块发现该文件（Caché 已经安装），则此步骤将不会运行。

好了，全部设置完毕后，我们可以运行安装了。

```
$ ansible-playbook dbserver.yml
PLAY [dbservers] *****
GATHERING FACTS *****
ok: [db1]
TASK: [hs_server_common | include_vars healthshare2015.yml] *****
ok: [db1]
TASK: [hs_server_common | include_vars os-RedHat.yml] *****
ok: [db1]
etc
etc
etc
TASK: [hs_server_common | Create default cache group] *****
changed: [db1]
TASK: [hs_server_common | Create default cache manager group] *****
changed: [db1]
TASK: [hs_server_common | Create default cache user] *****
changed: [db1]
TASK: [hs_server_common | Create default cache system users] *****
changed: [db1]
TASK: [hs_server_common | Create full hs install temp directory] *****
changed: [db1]
TASK: [hs_server_common | Check tar file (gunzipped already) does not exist] ***
ok: [db1]
TASK: [hs_server_common | Copy healthshare install file] *****
changed: [db1]
TASK: [hs_server_common | un zip hs folder] *****
changed: [db1]
TASK: [hs_server_common | un tar hs install] *****
changed: [db1]
TASK: [hs_server_common | Create hs install directory] *****
changed: [db1]
TASK: [hs_server_common | touch ztrak.conf.] *****
changed: [db1]
TASK: [hs_server_common | Process parameters file] *****
changed: [db1]
TASK: [hs_server_common | unattended install of hs using cinstall_silent] *****
changed: [db1]
TASK: [hs_server_common | copy hs key] *****
changed: [db1]
TASK: [hs_server_common | Set default hs instance] *****
changed: [db1]
TASK: [hs_server_common | restart apache to initialize CSP.ini file] *****
changed: [db1]
NOTIFIED: [hs_server_common | restart healthshare] *****
changed: [db1]
PLAY RECAP *****
db1 : ok=32 changed=21 unreachable=0 failed=0
```

如果我们查看目标服务器 — 数据库服务器 Caché 现在已经启动并运行。

```
$ ccontrol list
```

```
Configuration 'H2015' (default)
```

```
directory: /test/hs2015
versionid: 2015.2.1.705.0
conf file: cache.cpf (SuperServer port = 1972, WebServer = 57772)
status: running, since Wed Feb 17 15:59:11 2016
state: ok
```

总结

在后续的帖子中，我将构建包含其他任务的脚本，如编辑配置文件和使用 %installer 类配置应用程序。

如果您对此感兴趣并开始创建您自己的部署，请随时与我联系，提出问题或建议。

我经常在全球峰会上发表关于虚拟化和性能的演讲

因此，如果您参加今年的全球峰会，请介绍一下您自己，我非常乐意和您聊一聊的使用经验或任何其他系统架构话题。

-
Ansible

[#开源](#) [#系统管理](#) [#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E4%BD%BF%E7%94%A8-ansible-%E8%87%AA%E5%8A%A8%E5%8C%96%E9%83%A8%E7%BD%B2-cach%C3%A9-%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F-%E7%AC%AC-1>