VMware vRealize Application Services Migration Tool 1.1

User Guide

TECHNICAL WHITE PAPER

MAY 2017

VERSION 1.1

vmware®

Contents

Revision History

Date	Version	Description
January 2017	1.0	Initial version for vRealize Automation 7.1.
May 2017	1.1	Updated for vRealize Automation 7.3.

Migration Tool User Guide

This document provides guidance for using the vRealize Application Services Migration Tool.

Intended Audience

This guide is for administrators of vRealize Application Services, formerly vCloud Application Director, and vRealize Automation who want to migrate the application blueprints and deployment profiles from vRealize Application Services 6.2.x to vRealize Automation 7.3.

Introduction

The vRealize Application Services Migration Tool converts vRealize Application Services data into a format that can be imported into vRealize Automation 7.3.

The migration tool converts the services, external services, and application components used in a vRealize Application Services application to software in vRealize Automation. The tool then migrates the properties and scripts to the corresponding software.

The tool also converts application services blueprints and deployment profiles into vRealize Automation composite blueprints that contain the required software as well as dependencies between the software and property expressions. In addition to data conversion, the tool warns if any property expressions in an application blueprint cannot be migrated to a vRealize Automation composite blueprint property expression or requires a workaround.

Migration Tool Scope

The tool migrates these vRealize Application Services objects.

- Application
- Application version
- Blueprints
- Application component
- Services
- External services
- Deployment profiles with vCAC Cloud Provider Type

The tool does not migrate these vRealize Application Services objects.

- Artifacts
- Artifact repository
- Tags
- Operating systems
- Tasks
- Policies
- Logical templates
- Templates with pre-installed services
- Cloud provider
- Deployment environment
- Global Properties at deployment environment
- Update, rollback, and teardown profiles
- Deployment profiles with EC2 (Amazon) Cloud Provider Type
- Deployment profiles with vCloud 5.1.2-5.5 Cloud Provider Type

Software Prerequisites

Verify that you have these software resources in your source environment:

- vRealize Automation 6.2.x
- vRealize Application Services 6.2.x with an Enterprise Edition license

Verify that you have these software resources in your target environment:

- vRealize Automation 7.3
- vRealize CloudClient 4.2 or greater

Verify that you have downloaded these utilities

- Application Services CLI, `darwin-cli.jar`
  For information, see Start the CLI Remotely in the VMware vRealize Automation 6.2 Documentation Center.
- Migration CLI, `migration-cli.jar`
- vRealize CloudClient
  For information, see vRealize CloudClient 4.2 or greater.

Migration Procedures

Perform these procedures in the following order to migrate your application blueprints and deployment profiles from vRealize Application Services 6.2.x to vRealize Automation 7.3.

1. Export vRealize Application Services Application to an XML or ZIP File
   Use the Application Services CLI to export as an XML or ZIP file each vRealize Application Services application in the vRealize Automation 6.2.x environment.
2. Upgrade vRealize Automation 6.2.x Environment to 7.3
   Perform an in-place or side-by-side upgrade of vRealize Automation 6.2.x to 7.3.
3. Convert Application Services Content to a vRealize Automation ZIP File Using the Migration Tool
   Use the migration tool to convert each vRealize Application Services application XML or ZIP file to a ZIP file that you use to import the application to the vRealize Automation 7.3 target environment.
4. Download Software Component Files Stored on darwin.server or darwin.content.server to an HTTP Content Server
   Download the files used by software components that are hosted on darwin.server or darwin.content.server to an HTTP content server that is reachable by the upgraded vRealize Automation 7.3 environment.
5. Import the ZIP File to the vRealize Automation 7.3 Target Environment
   Use CloudClient to import each vRealize Application Services application to the target vRealize Automation 7.3 environment as a ZIP file.
6. Complete these post-import tasks
   Complete these tasks after import is finished.
7. Manually Resolve the Problems Identified by the Migration Tool Before Provisioning the Composite Blueprint
   The migration tool processes the input file during migration and generates a warning message if a property needs to be modified after importing the blueprint and software to vRealize Automation 7.3.

Export vRealize Application Services Application to an XML or ZIP File

Use the Application Services CLI to export as an XML or ZIP file each vRealize Application Services application in the vRealize Automation 6.2.x environment.

Prerequisite

Download the Application Services CLI, `darwin-cli.jar`, to the machine where you want to import the applications, for example, your local machine. For related information, see Using the CLI Export Command.

Procedure

1. On the machine where you downloaded `darwin-cli.jar`, start the Application Services CLI.
2. At the Application Services CLI command prompt, run the following command string to log in to the source environment.

   ```
   login --serverUrl https://serverFQDN:8443/darwin --username
   username.admin@sqa.local --password password --tenantId qe
   ```

   where

   - `https://serverFQDN:8443/darwin` is the URL of the vRealize Application Services server.
   - *`username.admin@sqa.local`* is the user name of a vRealize Application Services administrator with business group manager privileges.
   - *`password`* is the administrator password.
   - *`qe`* is the name of the vRealize Application Services tenant.

3. At the Application Services CLI command prompt, run the following command string to export each vRealize Application Services application as an XML or ZIP file.

   ```
   export-package --exportFilePath /tmp/filename.xml
   --fromGroup business group name
   --applicationVersion "application_name:version_number" --uncompressed
   --substituteSecuredProperties true
   ```

   where

   - `/tmp/`*`filename`*`.xml` is the path to the location of the application XML or ZIP file you are exporting.
   - *`business group name`* is the name of the business group that owns the application.
   - "*`application_name:version_number`*" is the name and version number of the application you are exporting.
   - `uncompressed` is an optional parameter. When present, `uncompressed` indicates the vRealize Application Services file is not a ZIP file.
   - `true` is the required value. `substituteSecuredProperties` must be set to `true` so that passwords are exported with a default value. If not set to `true`, the values of all secured properties are removed. If any passwords are a required property and exported without this option, the passwords will not have a value in the exported file and consequently the converted vRealize Automation ZIP file will not have a value for these passwords. This causes an error message to appear when you import the ZIP file with CloudClient.

Upgrade vRealize Automation 6.2.x Environment to 7.3

Perform an in-place or side-by-side upgrade of vRealize Automation 6.2.x to 7.3. After the vRealize Automation upgrade is complete, vRealize Automation Application Services is unavailable. The vRealize Automation appliance, however, remains available.

See Upgrading vRealize Automation 6.2.5 to 7.3 or Migrating vRealize Automation to 7.3.

Convert Application Services Content to a vRealize Automation ZIP File Using the Migration Tool

Use the migration tool to convert each vRealize Application Services application XML or ZIP file to a ZIP file that you use to import the application to the vRealize Automation 7.3 target environment.

Prerequisites

- Download the Migration CLI, `migration-cli.jar`, to the machine where you imported each vRealize Application Services application XML or ZIP file, for example, your local machine.
- Log in to the machine where you imported each vRealize Application Services application XML or ZIP file.

Procedure

1. Open a new command prompt and change directories to the location of the `migration-cli.jar` file.
2. Run the following command string to convert each application file to a ZIP file.

```
java -jar migration-cli.jar migrate --url=url
--username=user_name --password=password --tenant=tenant
--appdfile=file_path [--uncompressed=value]*
--outputdir=output_directory [--usemachineifreqd=value]*
[--debug=value]*
*optional
```

where

- `url` is the URL of the target vRealize Automation server.
- `user name` is a vRealize Automation user with Service Administrator and Catalog Administrator roles.
- `password` is the vRealize Automation user password.
- `tenant` is the vRealize Automation tenant.
- `file path` is path to the vRealize Application Services ZIP or XML file.
- `uncompressed` is an optional parameter. Set to `true` if the vRealize Application Services file is not a ZIP file.
- `Output directory` is the output file directory.

- `usemachineifreqd` is an optional parameter. Set the usemachineifreqd parameter to `true` to migrate information to a vRealize Automation blueprint if an Application Services blueprint contains any of the following properties:
  - Blueprint has a clustered node with cluster size more than 1.
  - Blueprint has storage disks configured on a blueprint node.
  - Deployment profile has vRealize Automation custom properties configured on a blueprint node.

  vRealize Automation uses references to blueprints selected in the deployment profile to represent a node in a vRealize Automation composite blueprint. Since these three properties cannot be configured with a reference to the original blueprint in the vRealize Automation composite blueprint, the migration tool copies the machine information from the original blueprint to a vSphere machine and adds these properties to it.

  If you do not use the usemachineifreqd parameter, the vRealize Automation composite blueprint uses a reference to the original blueprint instead of to a vSphere machine. The migration tool CLI also issues a warning message instructing you to make the corresponding changes to the original blueprint manually.
- `debug` is an optional parameter. If set to `true`, the log includes debug information. The log file, `appservices_migration.log`, is generated in the folder where `migration-cli.jar` file is run from.

Download Software Component Files Stored on darwin.server or darwin.content.server to an HTTP Content Server

Download the files used by software components that are hosted on darwin.server or darwin.content.server to an HTTP content server that is reachable by the upgraded vRealize Automation 7.3 environment. You can resolve many import issues after migration by pointing blueprints to the new content server.

Import the ZIP File to the vRealize Automation 7.3 Target Environment

Use CloudClient to import each vRealize Application Services application ZIP file to the target vRealize Automation 7.3 environment. For information about vRealize CloudClient, see https://code.vmware.com/tool/cloudclient.

Prerequisites

- Download CloudClient to the machine that contains each vRealize Application Services application ZIP file that you want to import to the vRealize Automation 7.3 target environment, for example, your local machine.
- Start CloudClient.

Procedure

1. At the CloudClient command prompt, run the following command string to log in to the vRealize Automation 7.3 environment.

   ```
   vra login userpass --server URL --user user_name
   --tenant tenant
   vRA Password for user_name: **********
   ```

   where

   - `URL` is the URL of the target vRealize Automation server.
   - `user name` is the user name of a vRealize Automation administrator with Application Architect and Software Architect role in the vRealize Automation 7.3 environment.
   - `tenant` is the target location for the imported files.
   - `**********` is the password for the vRealize Automation administrator.

2. At the Cloud Client command prompt, run the following command string to import each vRealize Application Services application ZIP file.

   ```
   vra content import –resolution VALUE --path /tmp/filename.zip
   ```

   where

   - `VALUE` determines how identical files are resolved. Set value to OVERWRITE or SKIP.
   - `/tmp/filename.zip` is the directory and the file name of the application you are importing.

3. At the command prompt, run the following command to show which blueprints and services are in a published state in vRealize Automation.

   ```
   vra content list
   ```

Complete these post-import tasks

After import, the vRealize Application Services templates in vSphere should be updated with a new agent from vRealize Automation 7.3. The vRealize Automation blueprints upgraded from 6.2.x to 7.3 continue to use the same vRealize Application services template, so no change is required for vRealize Automation. See Preparing to Provision Machines with Software.

Complete these tasks after import is finished.

- Change the passwords for the secured properties because properties are imported with a default password.
- Manually convert vRealize Automation services that represent vRealize Application services external services to a XaaS blueprint and replace these services using this XaaS blueprint in the vRealize Automation 7.3 migrated blueprint.

Manually Resolve the Problems Identified by the Migration Tool Before Provisioning the Composite Blueprint

The migration tool processes the input file during migration and generates a warning message in the log file similar to the following if a property needs to be modified after importing the blueprint and software to vRealize Automation 7.3:

```
**WARNING** Property 'HOSTNAME' in component 'sendNotification' uses 'hostname'
expression which is not available in vRA7. A workaround is to convert this
property to computed and use relevant command to find the hostname in the
script and set the resulting value to this computed property
```

The following sections describe situations where the migration tool issues a warning message.

Scripts

The BEANSHELL script in services and external services of Application Services is not supported for migration. It must be converted to one of the supported script types to use it in vRealize Automation 7.3.

Property Values

Some property values are not supported for migration in certain situations.

1. An array property with secured values, for example, an array of passwords, is supported in Application Services. An array property with secured values is not supported in vRealize Automation 7.3. The array is converted to an empty array after migration.
2. A computed property can have default values in Application Services. A computed property with default values is not supported in vRealize Automation 7.3. The values for this property must be set manually in the install, configure, or start scripts.

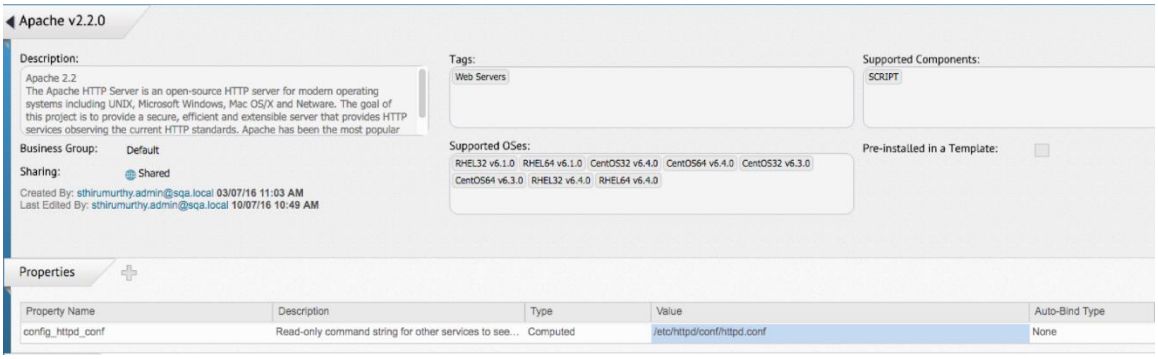

Figure 1 Property with a Computed Value

3. A content property can use the expression ${darwin.content.server.ip}, which refers to the Application Services appliance. The ${darwin.content.server.ip} property is seen in services, external services, and application components within the included application in Application Services and is specified in the path of the software content. In vRealize Automation 7.3, the content must be hosted in a content server, and the host name or IP address of the content server must replace ${darwin.content.server.ip}.

Figure 2 Content Property with ${darwin.content.server.ip}

4. A content property can also use the expression ${darwin.server.ip}, which refers to the Application Services appliance. This property is seen in included services and external services in Application Services and is specified in the path of the `darwin_global.conf` file, which defines the proxy configuration. The `darwin_global.conf` file is not available in vRealize Automation 7.3. To configure proxy information in vRealize Automation, you use the custom properties software.http.proxyHost, software.http.proxyPort, software.http.proxyUser, or software.http.proxyPassword. You might also need to delete the global_conf property name from the software components and comment out references to this property in the script.

Figure 3 Content Property with ${darwin.server.ip}

4. The host name property on a node in a blueprint is not supported for migration.

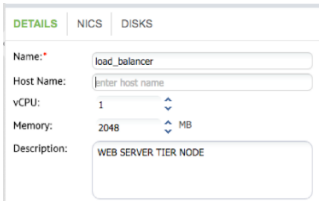

Figure 4 Host Name Property on a Node

Property Expressions

The following expressions used in an Application Services blueprint are not supported for migration.

1. An expression referring to the host name of a provisioned machine such as *self:hostname or appserver:hostname*, where “appserver” is the name of the target node, is not supported in vRealize Automation 7.3. You can resolve these problems in the following ways:

    - Convert self:hostname to a computed property, and in the install, configure, or start script, use a command like “hostname” to set the value to the computed property.

    - Create a new computed property for appserver:hostname, on the software of the appserver node, where appserver is the name of the target node. For example, use hostnameprop on software1 and assign the value in the install, start, or configure script. The existing property will refer to this computed property, for example, __resource~software1~hostnameprop.

2. An expression referring to the array index of a provisioned machine with a cluster like *self:node_array_index or appserver:node_array_index* , where appserver is the name of the target node, is not supported in vRealize Automation 7.3. You can resolve these problems in the following ways:

    - Convert *self:node_array_index* to a computed property, and in the install, configure, or start script, set the value $__cluster_index to the computed property.
    - Create a new computed property for *appserver:node_array_index* on the software of the appserver node. For example, use indexprop on software1 and assign the value $__cluster_index in the install, start, or configure script. The existing property refers to this computed property, for example, __resource~software1~indexprop.

Figure 5 Property with self:node_array_index

3. An expression referring to the array index of a provisioned machine in a cluster such as *self:disk_layout_info or appserver:disk_layout_info* is not supported in vRealize Automation 7.3. You can substitute specific storage related expressions like vSphereMachine~disks~capacity, vSphereMachine~disks~initial_location, or vSphereMachine~disks~storage_reservation_policy.

Figure 6 Property with self:disk_layout_info

4. An expression referring to a specific NIC of a provisioned machine like *self:NIC0_ip* or *self:NIC1_ip* is not supported in vRealize Automation 7.3

A template that is used in a blueprint creates another problem to solve. When you migrate your vRealize Automation 6.2.x environment, the virtual appliance FQDN might change. In this case, you must prepare a new template with scripts downloaded from the vRealize Automation 7.3 virtual appliance. The new template might need Java 1.6 and 1.8. Although you would typically use 1.8, your application might require 1.6.

For information and examples of blueprints and templates, see External Preparations for Provisioning in the VMware vRealize Automation Information Center.

To check for a more recent version of this guide, see VMware vRealize Automation Information.

You can find the most up-to-date technical documentation on the VMware Web site at: http://www.vmware.com/support.html The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to docfeedback@vmware.com.

VMware, Inc
3401 Hillview Ave
Palo Alto, CA 94304
www.vmware.com