MIGRATION GUIDE

CloudBolt software

VMware vRealize Automation Migration Guide

Reduce vRA Cost of Ownership and Eliminate Proprietary Code

Table of Contents

First Impressions of vRealize Automation 8 (vRA8)

Many of us have been eager to get our hands on vRealize Automation 8 (vRA8) for some time now. Maybe you've dabbled in the Hands On Labs with vRealize Automation Cloud (vRAC) for the past year. You've likely been through quite a few versions, upgrades, migrations, and code changes in your time working with the vRealize products. This article covers some of our first impressions of vRA8 and we're providing this as you weigh your options with the end-of-support for vRA7 in 2022.

Introduction

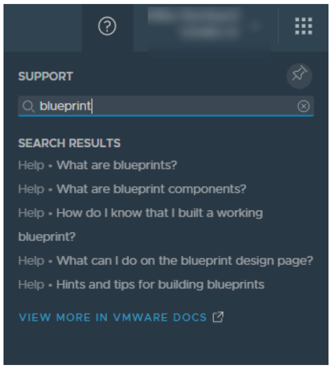
There is *a lot* to cover here. The product has been re-written from the ground up. The Cafe and IaaS nodes are no longer. Any vRO7 custom workflows will most likely have to be re-written. Migrations will have to be planned and extensive testing required prior to any production deployment. However, you are probably actively engaged in working with vRA in some fashion, and you're excited to see the new capabilities offered in vRA 8.x.

The following is a two-part summary of some of the very high-level aspects of the new system that have jumped out after working with VMware's reconstructed code for the vRA platform. In part one, we look at working with the documentation of vRA8 and tagging, the new way to manage your machine metadata. Part 2 will cover ABX and Policies.

Documentation and Community Backing

Learning a new platform and a new application architecture always come with challenges, and often those challenges relate to how well the product is documented. To get off the ground with vRA8, following the process of preparing infrastructure for resource deployment is easy. That portion of the setup is completed in about 30 minutes, and for the most part, it's clear to understand what's going on with the new components (Cloud Zones, Flavor Mappings, Image Mappings, etc.). Where you may hit some difficulties is in going through building the first blueprint and making it work with other constructs in the environment. YAML may be a new concept for some, and, while we found that the editor was intuitive, when we wanted to do something outside of a basic VM deployment, the documentation was lacking advanced use case examples.

The documentation is available right from the portal, which makes having to reference something much easier than in past versions. You can select the link to take you right to the VMware docs page. Just be careful, because that link will display search results for all VMware products, although these results can be further filtered to include just vRA8.



SUPPORT

blueprint

SEARCH RESULTS

Help • What are blueprints?

Help • What are blueprint components?

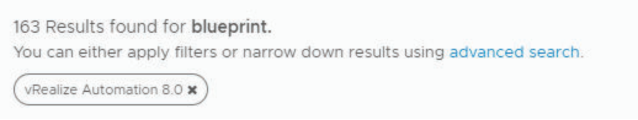
Help • How do I know that I built a working blueprint?

Help • What can I do on the blueprint design page?

Help • Hints and tips for building blueprints

VIEW MORE IN VMWARE DOCS

It doesn't appear that all the search results are available in the portal either. Results for "blueprint" above (5 results) showed 163 results on the official documentation site.



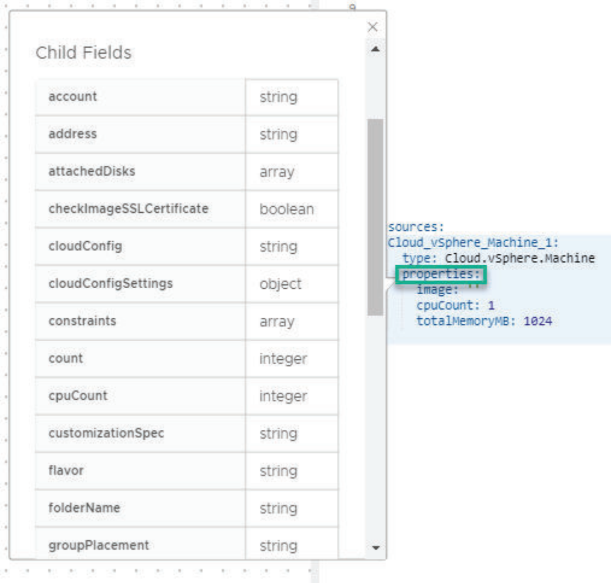
163 Results found for **blueprint.**

You can either apply filters or narrow down results using advanced search.

vRealize Automation 8.0 ✖

Introducing the vRA8 YAML Editor

When creating blueprints in past versions of vRA, the vRA Custom Properties Reference was an invaluable resource in showcasing the available properties for use in a blueprint. Sure, there are some "hidden" properties that never made it into the reference, but the community often filled in those gaps. In vRA8, there are multiple types of blueprint components, each translates to a different platform, and each of them has different available properties. One of the things that vRA8 executes well on so far is the ability to view the available properties for a given type directly from the YAML editor in the blueprint designer.



Child Fields

account	string
address	string
attachedDisks	array
checkImageSSLCertificate	boolean
cloudConfig	string
cloudConfigSettings	object
constraints	array
count	integer
cpuCount	integer
customizationSpec	string
flavor	string
folderName	string
groupPlacement	string

```
sources:
Cloud_vSphere_Machine_1:
  type: Cloud.vSphere.Machine
  properties:
    image: ''
    cpuCount: 1
    totalMemoryMB: 1024
```

However, what would be immensely helpful is a reference document of available properties as they pertain to each component type, as well as what each of these properties does and their expected inputs. These properties should be documented somewhere for reference away from the blueprint design canvas.

Brand New Code: Brand New Community Resources

In addition to the vendor-provided documentation, almost as important as the official product documentation are the available community resources for a product. The community backing and backlog of available resources for vRA7 is astounding, however vRA8 community articles are beginning to catch up.

Many of the articles may refer to vRealize Automation Cloud (vRAC), which is slightly more mature due to its accelerated release cadence and may or may not apply to all use cases of the on-prem vRA 8 product, e.g. tenancy and installation.

Tagging, Custom Properties, and Propagation

With vRA 8 comes a new way to manage your machine metadata. In the new platform, there are tags, capability tags, constraint tags, standard tags, project tags, project constraint tags, and custom properties. There is a bit of tag overload going on here. Some of these tags are used in placement decisioning, some of them propagate from higher level objects and go on to set tags on the endpoint itself, and some of them do not. According to the VMware documentation, each of the referenced tag types are defined as follows:

- Capability Tags: Enable you to define placement logic for deployment of infrastructure components.
- Constraint tags: You add constraint tags to blueprints and various other components within vRealize Automation Cloud Assembly to match capabilities defined on resources, cloud zones, and profiles to generate appropriate deployments.
- Standard tags: vRealize Automation Cloud Assembly applies standard tags to some deployments to support analysis, monitoring, and grouping of deployed resources. Note: although these are documented, they cannot be found whether in the GUI or the API. The overall intent behind them is unclear, or where they may actually be able to be referenced since they are not readily accessible in either the API or the GUI.
- Project Resource Tags: A project resource tag operates as a standardized identifying tag that you can use to manage the deployed resources and ensure compliance.
- Project Constraint Tags: A project constraint operates as a governance definition. It is a key:value tag that defines what resources the deployment request consumes or avoids in the project cloud zones.

A Change to Custom Properties

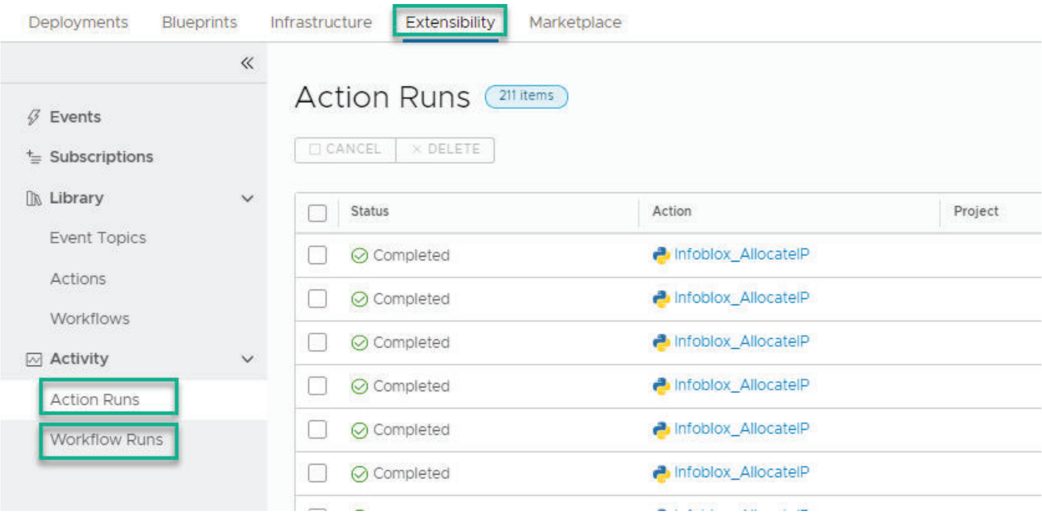
In custom properties are not like they were in vRA7. Custom properties can still be applied to a resource to specify metadata about that resource, but they are noticeably missing from infrastructure endpoints, and appear to be limited in total functionality compared to vRA7.

The only tags/custom properties/metadata that propagates down to deployed resources — such as virtual machines — appear to be project resource tags and tags set on the resource itself in the blueprint. We have worked with hundreds of customers who have realized simplified provisioning by being able to set Custom Properties directly where they were most applicable. Without the capability to specify metadata on the endpoints (cluster, vCenter, storage, network, etc.) and to have that metadata propagated during deployment, it can potentially lead to increased blueprint complexity, blueprint sprawl, and an increase in technical debt. However, with the OneFuse Property Toolkit (available for free with OneFuse Community Edition) it is possible to store and reference external metadata in OneFuse Property Sets, which can combat blueprint complexity, sprawl and mult-tool scenarios.

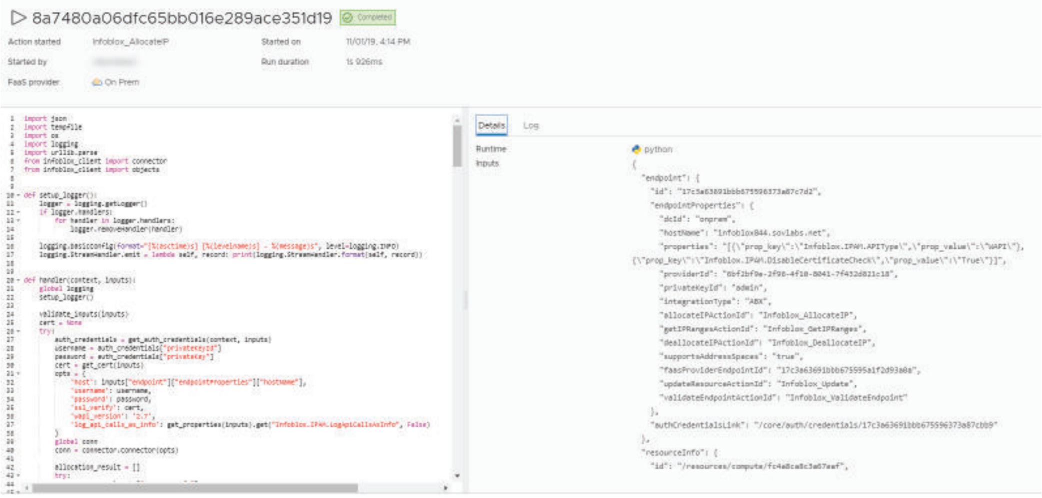
Action Based Extensibility (ABX), vRealize Orchestrator (vRO) and Extensibility

ABX is the new extensibility offering packaged with vRA 8 (in addition to vRealize Orchestrator) that uses a FaaS provider (AWS, Azure and On Prem offered today) to provide extensibility for the new platform. ABX is probably one of the more intriguing announcements surrounding vRA8 in our estimation.

ABX action runs show you the code that was used as well as the payload that was passed for use in the action. In fact, both Orchestrator and ABX runs can be monitored directly from the Cloud Assembly UI. No more having to log in to a separate interface to see what is happening with your extensibility. This methodology makes troubleshooting much more accessible.



	Status	Action	Project
□	Completed	Infoblox_AllocateIP	
□	Completed	Infoblox_AllocateIP	
□	Completed	Infoblox_AllocateIP	
□	Completed	Infoblox_AllocateIP	
□	Completed	Infoblox_AllocateIP	
□	Completed	Infoblox_AllocateIP	



Action started	Infoblox_AllocateIP	Started on	11/01/19, 4:14 PM
Started by	[redacted]	Run duration	1s 926ms
FaaS provider	On Prem		

```
import json
import tempfile
import os
import logging
import urllib.parse
from infoblox_client import connector
from infoblox_client import objects


def setup_logger():
    logger = logging.getLogger()
    if logger.handlers:
        for handler in logger.handlers:
            logger.removeHandler(handler)

    logging.basicConfig(format="[%(asctime)s] [%(levelname)s] - %(message)s", level=logging.INFO)
    logging.StreamHandler.emit = lambda self, record: print(logging.StreamHandler.format(self, record))


def handler(context, inputs):
    global logging
    setup_logger()

    validate_inputs(inputs)
    cert = None
    try:
        auth_credentials = get_auth_credentials(context, inputs)
        username = auth_credentials["privateKeyId"]
        password = auth_credentials["privateKey"]
        cert = get_cert(inputs)
        opts = {
            'host': inputs["endpoint"]["endpointProperties"]["hostName"],
            'username': username,
            'password': password,
            'ssl_verify': cert,
            'wapi_version': '2.7',
            'log_api_calls_as_info': get_properties(inputs).get("Infoblox.IPAM.LogApiCallsAsInfo", False)
        }
        global conn
        conn = connector.connector(opts)

        allocation_result = []
        try:
```

Runtime python

Inputs

```
{
  "endpoint": {
    "id": "17c3a63691bbb675596373a87c7d2",
    "endpointProperties": {
      "dcId": "onprem",
      "hostName": "infoblox844.sovlabs.net",
      "properties": "[{\"prop_key\":\"Infoblox.IPAM.APIType\",\"prop_value\":\"WAPI\"},
{\"prop_key\":\"Infoblox.IPAM.DisableCertificateCheck\",\"prop_value\":\"True\"}]",
      "providerId": "6bf2bf9e-2f98-4f18-8041-7f432d821c18",
      "privateKeyId": "admin",
      "integrationType": "ABX",
      "allocateIPActionId": "Infoblox_AllocateIP",
      "getIPRangesActionId": "Infoblox_GetIPRanges",
      "deallocateIPActionId": "Infoblox_DeallocateIP",
      "supportsAddressSpaces": "true",
      "faasProviderEndpointId": "17c3a63691bbb675595a1f2d93a8a",
      "updateResourceActionId": "Infoblox_Update",
      "validateEndpointActionId": "Infoblox_ValidateEndpoint"
    },
    "authCredentialsLink": "/core/auth/credentials/17c3a63691bbb675596373a87cbb9"
  },
  "resourceInfo": {
    "id": "/resources/compute/fc4e8ca8c3a67eef",
```

ABX actions and action flows (a way of chaining actions together like a workflow) are certainly learnable; however, compared to vRO, there are not yet as many samples to help ease the learning curve.

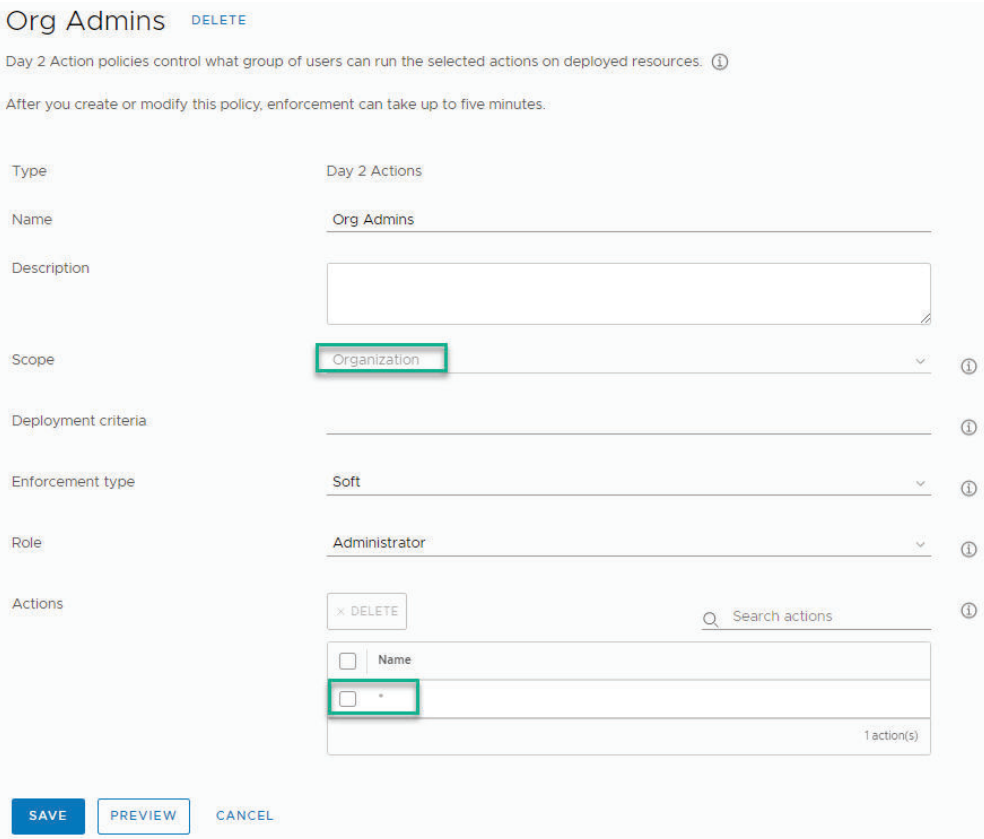
The implications that this change could have on your migrations as a customer could be quite staggering. If you have written a ton of complex extensibility code that centered upon using the vRA plug-in for vRO, you're going to significant rework.

vRealize Orchestrator 8 has removed the Java client entirely and has migrated entirely to a web interface like what was available in vRA7.6. You can now view and organize workflows and other content based on folder structure (as in vRA7) or by tags.

Entitlements and Policy Definitions

The entitlement constructs of past versions of vRA7 are now gone, and they have been replaced by policies in Service Broker in vRA8. These policies are in many ways more flexible, but also remove some of the granular permissioning that was present in vRA7 entitlements.

Organization-Level Policies: In vRA 6.x/7.x, some customers have spent a considerable amount of time writing custom workflows to copy entitlements from a "template" business group out to all the other (numerous) business groups. We'll cover one of these implementations now to illustrate the differences in the entitlement/permissions. This particular customer used template business groups, which have separate entitlements, each one pertaining to a different function (administrative, service account oriented, or end-user). These entitlements can easily number in the hundreds given there were also over a hundred business groups, and would need to match the corresponding entitlement tied to the template business group. In this case, if you wanted to add a single action that is entitled to all business groups, you would have to first entitle that action in the template business group, and then run a workflow that copied the entitlements out to the other business groups. In vRA8, it appears that this process has been addressed with the ability to add a policy at the organization (rather than project) level. With the policy in the following screenshot you can specify that all project administrators have access to all 2nd-day actions in the system within their respective projects. You probably wouldn't want to do this in a production environment, but it showcases some of the added flexibility inherent in vRA8 policies.



Org Admins DELETE

Day 2 Action policies control what group of users can run the selected actions on deployed resources. ⓘ

After you create or modify this policy, enforcement can take up to five minutes.

Type	Day 2 Actions
Name	Org Admins
Description	
Scope	Organization
Deployment criteria	
Enforcement type	Soft
Role	Administrator
Actions	× DELETE · Search actions

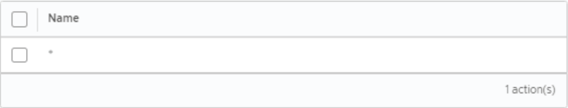
☐	Name
☐	*

1 action(s)

SAVE PREVIEW CANCEL

Wildcard Policy Selection: In past versions of vRA 7, you didn't have the option to specify, for example, that your administrative group be entitled to all machine level actions. Instead, you had to individually select each of the 20+ actions you wanted to entitle to administrative users of the business group. In vRA8, policies have a feature that allows you to specify a wildcard at any level of permissioning so that you can easily grant all permissions at the level required.

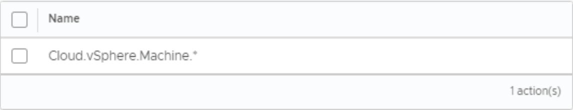
For example, to entitle every single action for every single type, you would simply select the '*' wildcard:



☐	Name
☐	*

1 action(s)

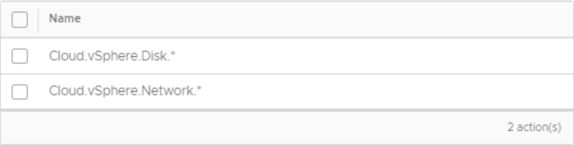
If you wanted to grant access to every vSphere level action, you would select Cloud.vSphere.Machine.*



☐	Name
☐	Cloud.vSphere.Machine.*

1 action(s)

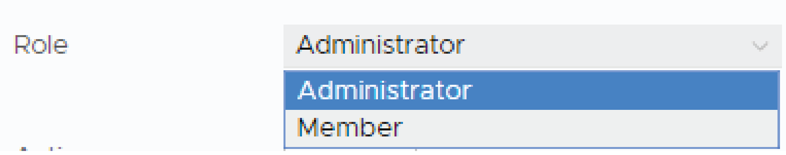
Or if you had a user that only needed access to actions for vSphere Disk and Network related actions, you could do the following:



☐	Name
☐	Cloud.vSphere.Disk.*
☐	Cloud.vSphere.Network.*

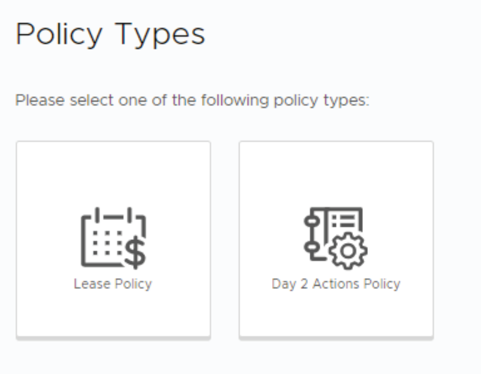
2 action(s)

Individual Group or User Policy Assignment: In the previous example with the template business group, we had those base level actions, but we also had requirements where we would create specialty entitlements within a business group to entitle specific actions above and beyond the base entitlements that all users would see. These specialty entitlements would then be assigned to individual users or a Security Group in the Business Group that held a subset of the users in the business group that were often not administrative level members. In vRA 8 policies, there is currently no mechanism for assigning granular permissions to a subset of the members or administrators in a project. Policies are applied to either Administrators or Members of the project.



Role: Administrator

Catalog Item Policies: When you go to create a new policy, you may notice that the capability to restrict access to Catalog Items in vRA 8 is no longer available. The only options that you are presented with are Lease Policy or a Day 2 Actions Policy.



Policy Types

Please select one of the following policy types:

It seems that Catalog Items are entitled to the project as a whole and there is no way to filter the available Catalog Items on an individual user basis. Quite a few of the customers we work with require the ability to restrict access to not only 2nd Day Actions, but to Catalog Items as well within a project.

VMware has made massive strides in many areas to improve upon user feedback from past versions. Cross-Cloud capabilities are markedly enhanced in the new version, and Infrastructure as Code is now a reality for vRA blueprints. Whether your first foray in the vRA8 world, or you've worked with vRA since the DynamicOps days, there is a learning curve ahead.

An Overview of the vRA8 Migration Assessment Tool

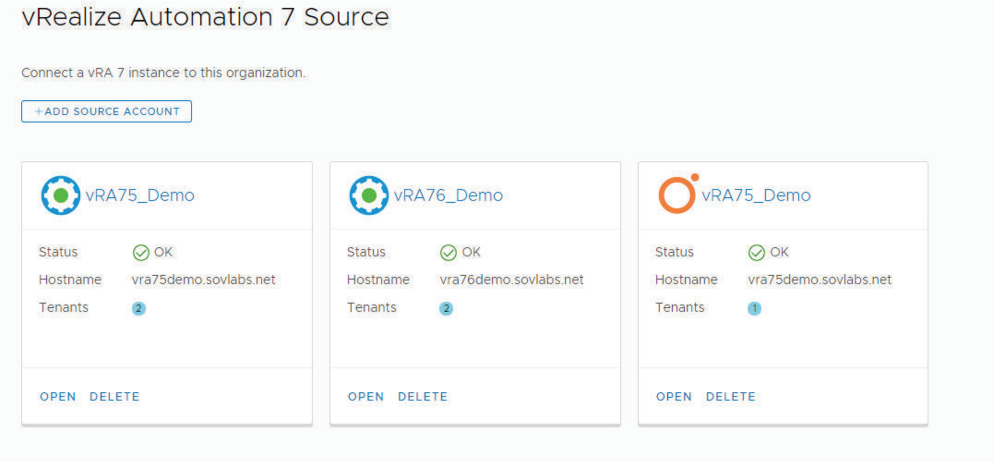
vRealize Automation veterans may still remember the migrations from vRA 5.x to 6.x and 6.x to 7.x. However, for many enterprises utilizing vRealize Automation, the migration from vRA 7.x to vRA 8.x will be their first major vRA migration. In an ideal world these migrations would only take a few clicks of the mouse.

Migrating from vRA7 to vRA8 is going to be a lot like switching banks. Trying to figure out all the services that have your bank card on file for automatic billing and moving over automatic bill payments is tedious and time consuming. Which payments are monthly, quarterly, annually, and for how much? Wouldn't it be nice if there were a simple tool to identify and update all the services that you have on auto-pay?

vRealize Automation is a lot like that. Whether it's frequently deployed common workloads, special purpose blueprints that are used a few times a month, quarterly, or a handful or fewer times a year, or software customizations specific to vRA, you must identify everything that will require handholding when you begin the migration from vRA7 to vRA8. The good news is that, unlike your bank, VMware does offer a migration assessment tool to help you determine if your blueprints are ready to be migrated from vRA7 to vRA8.

vRealize Automation Migration Assessment Tool

Think of the vRA assessment tool as an overview of your bank statements that you need to comb through. To start, you add an existing vRA7 instance to this tool so it can do a discovery on what you have and provide migration recommendations. The tool offers an option to validate an embedded vRO server. For testing, we configured a vRA 7.5 and a vRA 7.6 environment to see how they look for migration. We also chose to look at vRO in only one of my environments.



vRealize Automation 7 Source

Connect a vRA 7 instance to this organization.

vRA75_Demo

Status OK
Hostname vra75demo.sovlabs.net
Tenants 2

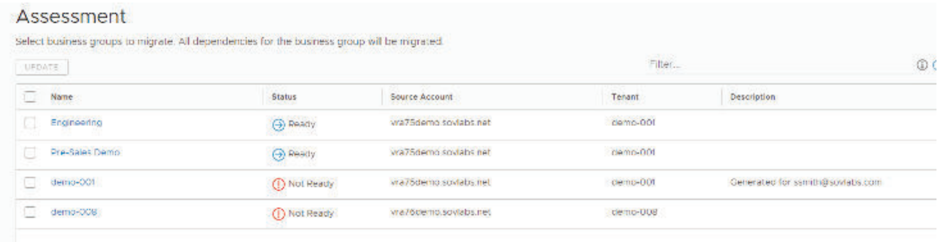
vRA76_Demo

Status OK
Hostname vra76demo.sovlabs.net
Tenants 2

vRA75_Demo

Status OK
Hostname vra75demo.sovlabs.net
Tenants 1

Once the assessment is complete, it will list all the business groups available for the tenants you have selected and give you a status for each. In the case (below) there are 4 business groups, but only 2 of them have any blueprints. Can you guess which two they are?



Assessment

Select business groups to migrate. All dependencies for the business group will be migrated.

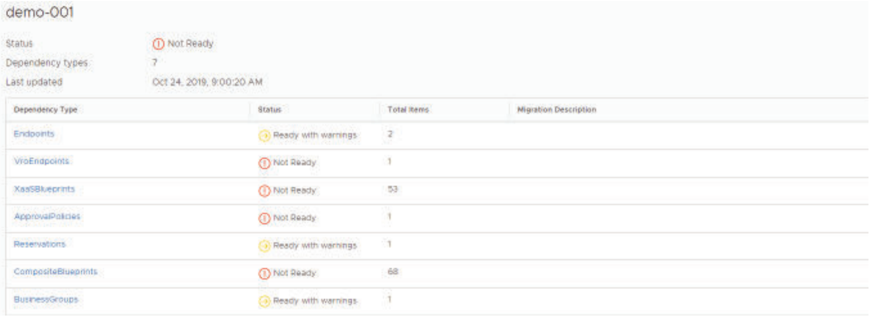
UPDATE

Filter...

☐	Name	Status	Source Account	Tenant	Description
☐	Engineering	Ready	vra75demo.sovlabs.net	demo-001	
☐	Pre-Sales Demo	Ready	vra75demo.sovlabs.net	demo-001	
☐	demo-001	Not Ready	vra75demo.sovlabs.net	demo-001	Generated for ssmith@sovlabs.com
☐	demo-008	Not Ready	vra76demo.sovlabs.net	demo-008	

At the top it states “Assessment,” then, just below that, it states “*Select business group to migrate. All dependencies for the business group will be migrated.*” Currently no migration is available.

Once you open the business group you want to work with, you are then presented with a list of dependencies for that business group. Dependencies include constructs like Endpoints, Reservations, Approval Policies, vRO Endpoints, XaaS Blueprints, and Composite Blueprints.



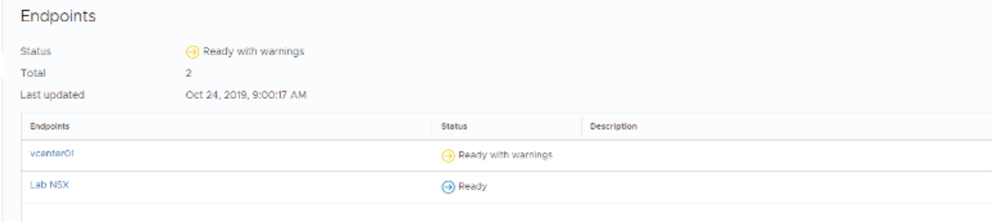
demo-001

Status: Not Ready
Dependency types: 7
Last updated: Oct 24, 2019, 9:00:20 AM

Dependency Type	Status	Total Items	Migration Description
Endpoints	Ready with warnings	2	
VroEndpoints	Not Ready	1	
XaaSBlueprints	Not Ready	53	
ApprovalPolicies	Not Ready	1	
Reservations	Ready with warnings	1	
CompositeBlueprints	Not Ready	68	
BusinessGroups	Ready with warnings	1	

Endpoints

Drilling down into Endpoints reveals a list of available Endpoints with a migration status in an environment. We have two Endpoints and one is Ready and one is Ready with Warnings.

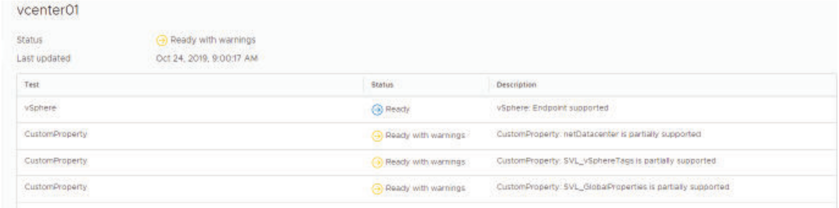


Endpoints

Status: Ready with warnings
Total: 2
Last updated: Oct 24, 2019, 9:00:17 AM

Endpoints	Status	Description
vcenter01	Ready with warnings	
Lab NSX	Ready	

Upon Upon further investigation into the endpoint with warnings, you see that the warnings are all related to custom properties defined on the EndPoint. This explanation makes sense considering properties are limited in vRA 8, and you cannot define properties on vRA8 Endpoints. What we find to be a little misleading is the migration assessment tool shows that the properties are “partially supported.”



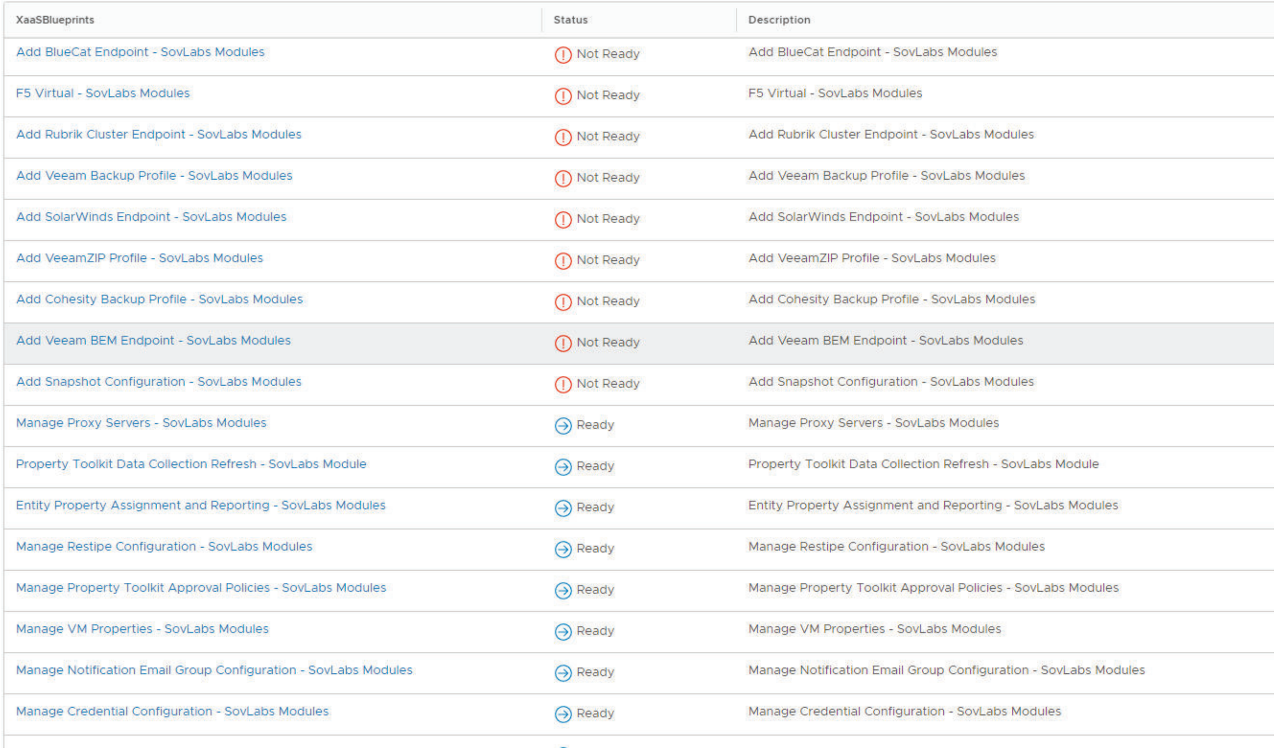
vcenter01

Status: Ready with warnings
Last updated: Oct 24, 2019, 9:00:17 AM

Test	Status	Description
vSphere	Ready	vSphere: Endpoint supported
CustomProperty	Ready with warnings	CustomProperty: netDatacenter is partially supported
CustomProperty	Ready with warnings	CustomProperty: SVL_vSphereTags is partially supported
CustomProperty	Ready with warnings	CustomProperty: SVL_GlobalProperties is partially supported

XaaS Blueprints

When investigating items under XaaS Blueprints (see image below), we see a large number of XaaS Blueprints that are not ready to migrate, and a handful that are ready to migrate. After looking through the list it seems that the determining factor between an XaaS Blueprint being ready or not is whether the blueprint uses vRO custom resources. That criteria is not a great way of determining and reporting if something is ready to migrate to vRA8. Furthermore, we reviewed the list of XaaS Blueprints that it says are ready and one by one we realized they would not work with vRA8.



XaaSBlueprints	Status	Description
Add BlueCat Endpoint - SovLabs Modules	Not Ready	Add BlueCat Endpoint - SovLabs Modules
F5 Virtual - SovLabs Modules	Not Ready	F5 Virtual - SovLabs Modules
Add Rubrik Cluster Endpoint - SovLabs Modules	Not Ready	Add Rubrik Cluster Endpoint - SovLabs Modules
Add Veeam Backup Profile - SovLabs Modules	Not Ready	Add Veeam Backup Profile - SovLabs Modules
Add SolarWinds Endpoint - SovLabs Modules	Not Ready	Add SolarWinds Endpoint - SovLabs Modules
Add VeeamZIP Profile - SovLabs Modules	Not Ready	Add VeeamZIP Profile - SovLabs Modules
Add Cohesity Backup Profile - SovLabs Modules	Not Ready	Add Cohesity Backup Profile - SovLabs Modules
Add Veeam BEM Endpoint - SovLabs Modules	Not Ready	Add Veeam BEM Endpoint - SovLabs Modules
Add Snapshot Configuration - SovLabs Modules	Not Ready	Add Snapshot Configuration - SovLabs Modules
Manage Proxy Servers - SovLabs Modules	Ready	Manage Proxy Servers - SovLabs Modules
Property Toolkit Data Collection Refresh - SovLabs Module	Ready	Property Toolkit Data Collection Refresh - SovLabs Module
Entity Property Assignment and Reporting - SovLabs Modules	Ready	Entity Property Assignment and Reporting - SovLabs Modules
Manage Restipe Configuration - SovLabs Modules	Ready	Manage Restipe Configuration - SovLabs Modules
Manage Property Toolkit Approval Policies - SovLabs Modules	Ready	Manage Property Toolkit Approval Policies - SovLabs Modules
Manage VM Properties - SovLabs Modules	Ready	Manage VM Properties - SovLabs Modules
Manage Notification Email Group Configuration - SovLabs Modules	Ready	Manage Notification Email Group Configuration - SovLabs Modules
Manage Credential Configuration - SovLabs Modules	Ready	Manage Credential Configuration - SovLabs Modules

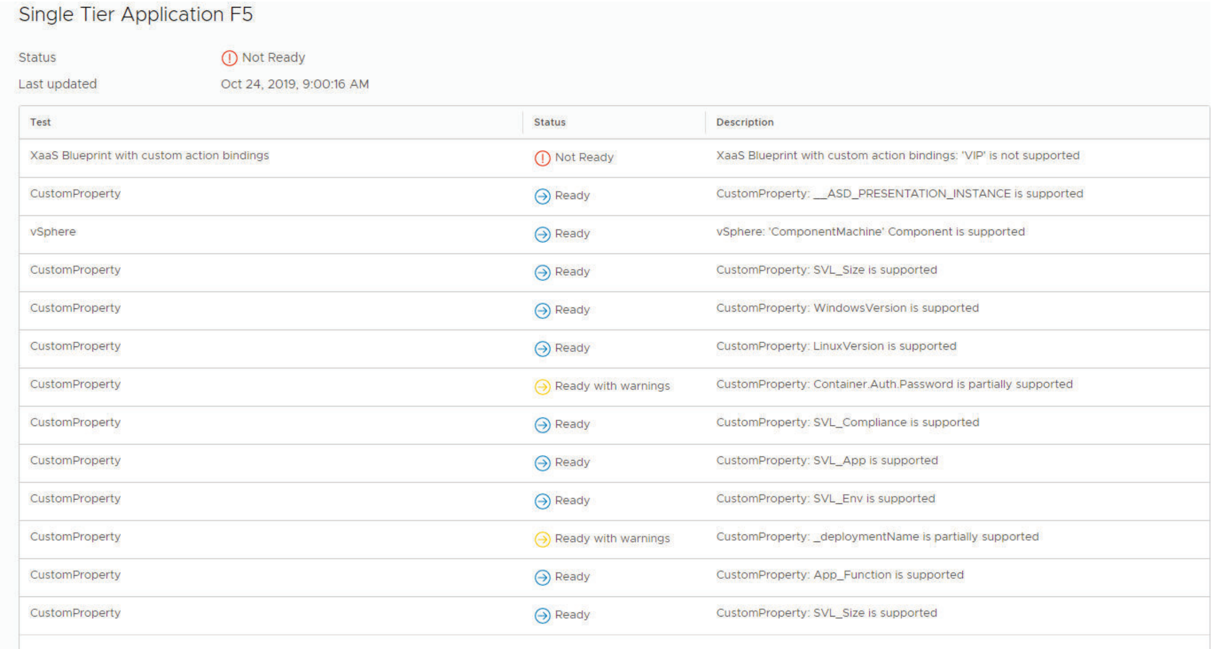
The assessment tool looks for a few very finite requirements that VMware knows are currently not available in vRA8. But that's just the tip of the iceberg. You will need to understand what your vRO workflow is doing that the XaaS Blueprint is calling. The assessment tool does not inspect your workflow and understand what its functions are.

Here's an example: one of the XaaS items that is ready for migration is used to inspect and report on vRA7 custom properties. It looks across all the constructs in vRA 7 to report on where properties are defined and what their values are. It even gives you the option to bulk update those properties. Although the assessment tool says this functionality is ready for migration, it certainly will not work on vRA8.

Composite Blueprints

When investigating the assessment tool for Composite Blueprints, the tool presented me with some blueprints that were *Not Ready*, some that are *Ready With Warnings*, and some that are *Ready*.

Here is what I discovered that each of those terms actually means.



Single Tier Application F5

Status	Not Ready
Last updated	Oct 24, 2019, 9:00:16 AM

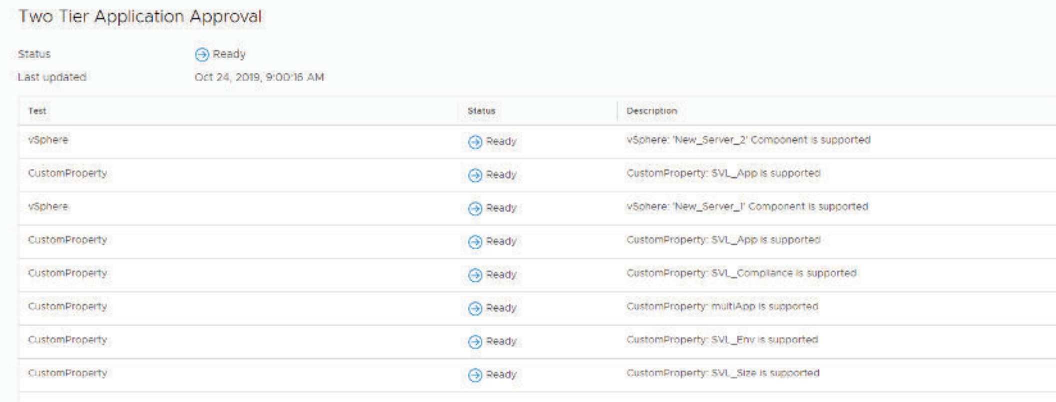
Test	Status	Description
XaaS Blueprint with custom action bindings	Not Ready	XaaS Blueprint with custom action bindings: 'VIP' is not supported
CustomProperty	Ready	CustomProperty: __ASD_PRESENTATION_INSTANCE is supported
vSphere	Ready	vSphere: 'ComponentMachine' Component is supported
CustomProperty	Ready	CustomProperty: SVL_Size is supported
CustomProperty	Ready	CustomProperty: WindowsVersion is supported
CustomProperty	Ready	CustomProperty: LinuxVersion is supported
CustomProperty	Ready with warnings	CustomProperty: Container.Auth.Password is partially supported
CustomProperty	Ready	CustomProperty: SVL_Compliance is supported
CustomProperty	Ready	CustomProperty: SVL_App is supported
CustomProperty	Ready	CustomProperty: SVL_Env is supported
CustomProperty	Ready with warnings	CustomProperty: _deploymentName is partially supported
CustomProperty	Ready	CustomProperty: App_Function is supported
CustomProperty	Ready	CustomProperty: SVL_Size is supported

Not Ready - Blueprints with nested blueprints inside, blueprints with items other than networks on the canvas, and Blueprints with XaaS items on the canvas.

Ready With Warnings - Blueprints with reserved vRA7 custom properties, including properties such as _deploymentName, Extensibility.LifeCycleProeprties, and other vRA7 specific reserved properties.

Ready - If the blueprint doesn't meet any of the criteria listed in the above two statuses, then the blueprint is deemed "Ready". Be careful though: the assessment tool has no idea what other properties values are actually used for. It only knows that they are not reserved vRA7 properties that have been deprecated, so they must be good.

There does not appear to be a check to see if these properties are used to trigger event subscriptions, which would potentially make them unsupported if the state they are triggering is no longer available in vRA8. Further, the assessment tool has no idea what they are being used for in your custom or third-party workflows, so it's best to assume that, if they are used in vRO workflows, they are not ready for vRA8.



Two Tier Application Approval

Status	Ready
Last updated	Oct 24, 2019, 9:00:16 AM

Test	Status	Description
vSphere	Ready	vSphere: 'New_Server_2' Component is supported
CustomProperty	Ready	CustomProperty: SVL_App is supported
vSphere	Ready	vSphere: 'New_Server_1' Component is supported
CustomProperty	Ready	CustomProperty: SVL_App is supported
CustomProperty	Ready	CustomProperty: SVL_Compliance is supported
CustomProperty	Ready	CustomProperty: multiApp is supported
CustomProperty	Ready	CustomProperty: SVL_Env is supported
CustomProperty	Ready	CustomProperty: SVL_Size is supported

Approval Policies, Reservations, Business Groups, and other Constructs

When looking at these other constructs, the assessment tool is really just looking at the custom properties defined on the constructs. If the assessment tool sees vRA7 reserved properties that are not supported within vRA8, the tool will flag these properties as “Ready with warnings.” Otherwise, these properties will be considered “Ready.”

The assessment tool may give you some insights into what you should look out for, but, in its current form, the tool’s output can be very misleading. The best approach to building a migration plan is to start with a manual assessment. When using third party integrations, custom integrations, and your own properties, the assessment tool just doesn’t have the insight to know if the properties are supported.

Also, keep in mind the differences in concepts and constructs between vRA7 and vRA8. These changes may be cause to rethink how you accomplished something in vRA7, and may justify a change in property locations.

In Part 1 of this two-part series on the vRealize Automation Migration Assessment Tool, we looked at the vRA8 migration tool to see how it might help you plan your migration from vRA7 to vRA8. In this article, we are going to look at what it will take to migrate your custom workflows from vRA7 to vRA8. To start, we will explore what a custom workflow looks like today in vRA7.

vRA7 Workflow Components

In vRA 7, there are several components that come together to make the magic happen. Each component plays a vital role in how you design, build, and invoke your customizations.

Event Broker

The event broker was introduced in vRA7 to make it easier to trigger the workflow stubs that existed within the IaaS server. These stubs were always there, but only a handful were accessible and they were not easy to configure. The Event Broker also introduced a more granular way to decide when a workflow should or should not be executed. Although the event broker has dozens of events you could subscribe to the following were the most commonly used:

- Machine Requested
- Building Machine
- Machine Provisioned
- Machine Activated
- Machine Destroyed

Many of these states included a pre- and a post-execution allowing you to decide if you wanted to execute your workflow before or after vRA’s execution of that state. These states that are the core of vRA7 extensibility no longer exist in vRA8.

vRA 8 has introduced new event subscription topics that — in the long term — could be very welcomed; however, in the short term, it will mean reworking all your existing workflow customizations. These new event subscription topics are listed below.

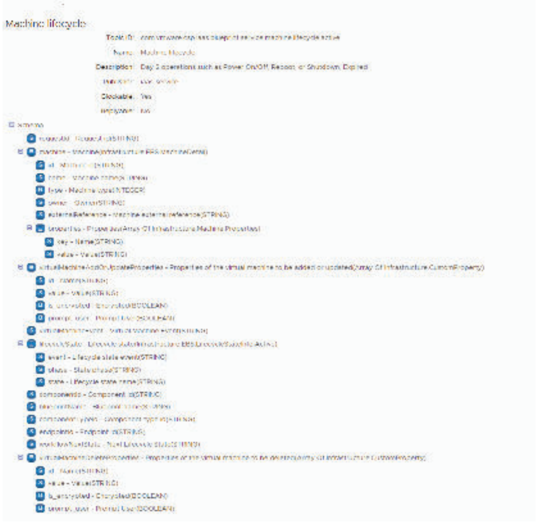
- Deployment Level
    - Deployment Requested
    - Compute Reservation
    - Compute Allocation
    - Network Configure
    - Deployment Completed
- Network Level
    - Deployment Resource Requested
    - Deployment Resource Completed
- Machine Level
    - Deployment Resource Requested
    - Compute Reservation
    - Deployment Requested
    - Compute Reservation

As you can see, this change is significant to the event subscriptions as well as the overall lifecycle that a virtual machine goes through when it is requested. This change alone is not so bad, right? Simply setup new subscriptions to execute your existing vRealize Orchestrator (vRO) workflows and all is good, right? Well, not exactly.

Event Schemas

In vRealize Automation, all of the Events that are part of the Event Broker have a schema. Looking at the schema for LifeCycle Events in vRA7, you will notice that it includes items like *machine*, *lifecyclestate*, *properties*, and *virtualmachineAddorUpdateProperties*. These items are all included in what is commonly known as the Payload. The payload is sent to vRO and you can then reference information about the request. You can access all the custom properties associated with the request under *machine.properties*.

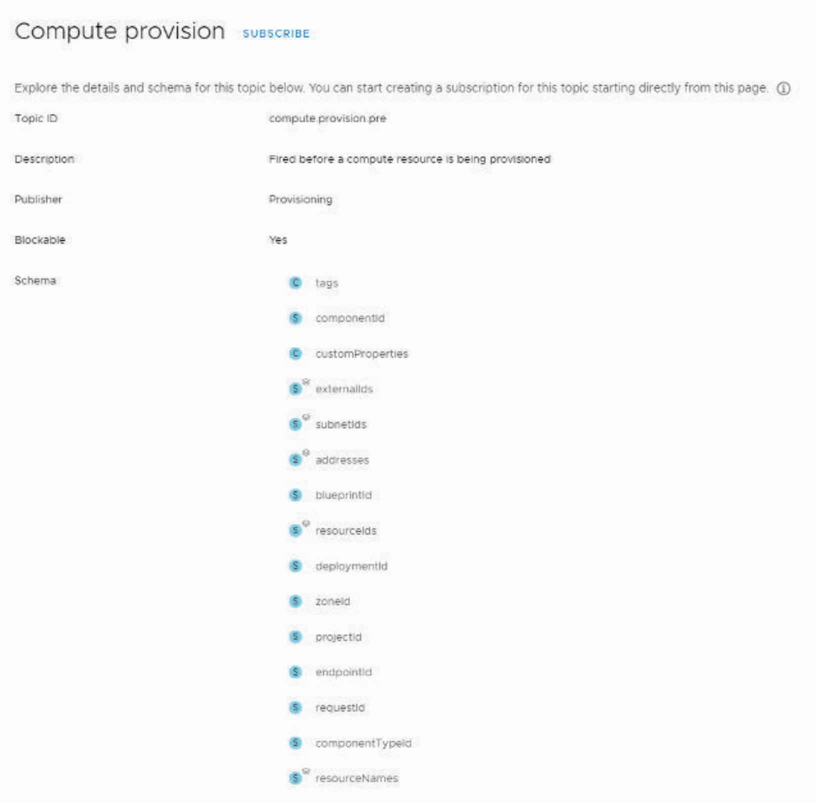
All workflows built using vRA7 use this payload to take in information such as properties and also write back updated information using the *virtualMachineAddorUpdateProperties* to impact the outcome of the request. The payload is sent by vRA. Then, you extract the information you are looking for using the schema below. That data is then used throughout your workflow.



```
Machine lifecycle
        Topic ID: com.vmware.csp.iaas.blueprint.service.machine.lifecycle.active
            Name: Machine lifecycle
     Description: Day 2 operations such as Power On/Off, Reboot, or Shutdown, Expired
       [illegible]: [illegible]
       Blockable: Yes
       Replyable: No
⊟ Schema
    requestId - Request id(STRING)
  ⊟ machine - Machine(Infrastructure.EBS.MachineEntity)
      id - Machine id(STRING)
      name - Machine name(STRING)
      type - Machine type(INTEGER)
      owner - Owner(STRING)
      externalReference - Machine external reference(STRING)
    ⊟ properties - Properties(Array Of Infrastructure.Machine.Properties)
        key - Name(STRING)
        value - Value(STRING)
  ⊟ virtualMachineAddOrUpdateProperties - Properties of the virtual machine to be added or updated(Array Of Infrastructure.CustomProperty)
      id - Name(STRING)
      value - Value(STRING)
      is_encrypted - Encrypted(BOOLEAN)
      prompt_user - Prompt User(BOOLEAN)
    virtualMachineEvent - Virtual Machine event(STRING)
  ⊟ lifecycleState - Lifecycle state(Infrastructure.EBS.LifecycleStateInfo.Active)
      event - Lifecycle state event(STRING)
      phase - State phase(STRING)
      state - Lifecycle state name(STRING)
    componentId - Component id(STRING)
    blueprintName - Blueprint name(STRING)
    componentTypeId - Component type id(STRING)
    endpointId - Endpoint id(STRING)
    workflowNextState - Next Lifecycle state(STRING)
  ⊟ virtualMachineDeleteProperties - Properties of the virtual machine to be deleted(Array Of Infrastructure.CustomProperty)
      id - Name(STRING)
      value - Value(STRING)
      is_encrypted - Encrypted(BOOLEAN)
      prompt_user - Prompt User(BOOLEAN)
```

In vRA8, this payload has changed significantly. In fact, it has changed so much that it's not even called *payload* anymore. It's new name is "*Inputs*". Not only has the name changed, but what is included has changed as well. If we start to look at the *machine.properties* item in vRA7, we know it included all the custom properties associated with the request, including inputs to blueprint properties, business group properties, property groups, reservations, compute resources, and endpoint properties. vRA8 uses a different methodology and has very limited support for properties. These similar constructs now use tags.

This change is a major shift in the architecture. Most workflows are built around these concepts and being able to acquire the needed information from the properties being defined in the appropriate locations. Because of this change, workflows built to run on vRA7 will not function on vRA8. Below is the schema from the "Compute Provision" state in vRA8. Notice the differences between this vRA8 schema (below) and the vRA7 schema (above).



Compute provision SUBSCRIBE

Explore the details and schema for this topic below. You can start creating a subscription for this topic starting directly from this page. ⓘ

Topic ID	compute.provision.pre
Description	Fired before a compute resource is being provisioned
Publisher	Provisioning
Blockable	Yes
Schema	C tags S componentId C customProperties S externalIds S subnetIds S addresses S blueprintId S resourceIds S deploymentId S zoneId S projectId S endpointId S requestId S componentTypeId S resourceNames

The vRA plugin for vRO

If you were building vRO workflows for vRA7, you are most likely using the vRA plugin for vRO. The plugin allows you to create endpoints for both the IaaS host as well as the Cafe host to be used in your workflows. The vRA7 plugin for vRO also has a number of prebuilt actions and workflows that you can use to make your life easier. As of vRA 8.4 and above, there is now a vRA plug-in for vRO. The plug-in sample content covers several use cases by abstracting vRA8/vRAC APIs including REST operations, Event Broker, Tag and Custom Properties, Deployment Day2 operations, Organization onboarding, Request/Approvals, Types conversion and Dynamic Types. Samples list: https://developer.vmware.com/samples?id=7650

There Is Another Option

I'm sure many of you were hoping that workflows that you build, are building, or have built for use in vRA 7 would be somewhat transferable to vRA 8. Unfortunately, it doesn't look like it will be that easy. To salvage your current workflows, you will need to consider the following questions.

Does your vRA7 workflow...

- rely on properties from vRA7 constructs to make decisions?
- rely on the vRA7 payload the schema associated with it?
- rely on the vRA plugin for vRO?

If you answered yes to even one of these questions, then you will have to rework your workflows for use in vRA8.

Or will you?!?

There is another option. We have been helping many vRA customers migrate away from their customizations on vRA7 using OneFuse. Whether you are solely using lifecycle workflows, or using XaaS Blueprints in vRA to meet your business needs, we can help.

For example, many customers decide to use vRA onboarding vs. using the vRA migration assistant due to the complexities of prerequisites from the assistant. For customers using onboarding, OneFuse provides a migration workflow that utilizes property sets to bulk import selective VMs and their metadata and can perform translations to property keys/values and add or omit any metadata – all without any custom code!

Here are several ways we help customers move beyond vRA7 while reducing tech debt, management overhead and complexity:

1. **Productized solution**: Less custom coding or services and overall lower costs, higher functionality, higher quality and faster turn on of more automation. Product support model, including roadmap and upgrades future proofs investment.

2. **Portability**: Re-use of integrations between multiple vRA components (code stream, vRO, Cloud Assembly machine lifecycle, custom resources, ABX) and multiple tools/platforms (vRA, Terraform, Ansible, ServiceNow, etc.) lowers overhead and reduces tech debt

3. **Security and compliance**: Improve security posture through template-based automation over custom scripts and workflows. Implement least privilege through a policy-based service approach that provides guard rails for use cases.

4. **Consumption model**: Improve standardization and consumption within and across automation tools, enabling new use cases to be delivered faster with more consistency and less complexity.

5. **Assessment tool**: Learn more about your environment and dependencies to help you plan the migration from vRA7 with confidence.

6. **Migration capabilities**: Migrate names, IPs, Property Groups, SovLabs or other configurations to OneFuse. Additionally, migrate and/or transform VM properties from vRA7 to vRA8.

7. **Blueprint consolidation / simplification**: Use OneFuse Property Sets to simplify vRA8/vRAC blueprints without complex blueprint expressions syntax (ECMAScript) allowing for dynamic keys, values, rich JSON support, nested