CloudBolt software | SovLabs

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. If any of you are like me, you may have dabbled in the Hands On Labs with vRealize Automation Cloud (formerly CAS) for the past year. But, without having the capability or the requisite login for the service, you kept abreast of any updates coming out of the vRAC camp. I have been through quite a few versions, upgrades, migrations and code changes in my time working with the vRealize products. Most of this experience was spent in architecture, engineering, and development at a large enterprise customer, where we had to carefully navigate each of these changes. This article covers some of my first impressions of vRA8 as an advanced user of the product since the early days, circa vRA6.

Introduction

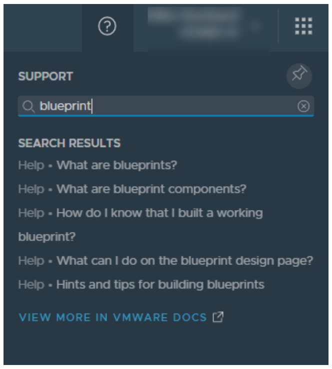
There is *a lot* to cover here. The product has been re-written from the ground up. The Cafe and IaaS nodes that we are all familiar with are now gone. Custom workflows will most likely have to be completely re-written. Migrations will have to be planned and extensive testing required prior to any production deployment. It will likely take the better part of the next year for the community to get a handle on the new product. However, if you're reading this blog post, you are probably actively engaged in working with vRA in some fashion, and you're excited to see the new capabilities offered in vRA8.x.

The following is a two-part summary of some of the very high level aspects of the new system that have jumped out at me in the first 2-3 weeks of working with VMware's reconstructed code for vRealize Automation. In part one, I explain my experience working with the Documentation of vRA8 and Tagging, the new way to manage your machine metadata. Part 2 will cover my take on ABX and Policies. Also, we're going in depth on these and other topics in our December 10th webinar: Top vRA8 Deployment Considerations. Click here to sign up.

Documentation and Community Backing

Learning a new platform and a new application architecture always come with challenges, and oftentimes those challenges relate to how well the product is documented. That being said, to get off the ground with vRA8, I was able to fairly easily follow the process of preparing my infrastructure for resource deployment. I completed that portion of the setup in about 30 minutes (not including the initial deployment) and, for the most part, clearly understood what was going on with the new components (Cloud Zones, Flavor Mappings, Image Mappings, etc.). Where I started to hit some difficulties was in going through building my first blueprint and making it work with other constructs in the environment. YAML is a new concept for me, and, while I found that the editor was intuitive, when I wanted to do something outside of a basic VM deployment, the documentation didn't have very many advanced use case examples.

The documentation is available right from the portal, which makes having to reference something much easier than in past versions. And, 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?

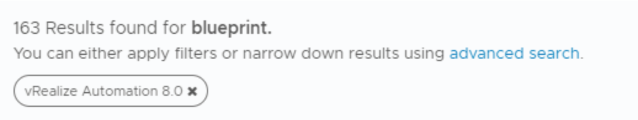
Help • What are blueprint components?

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

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

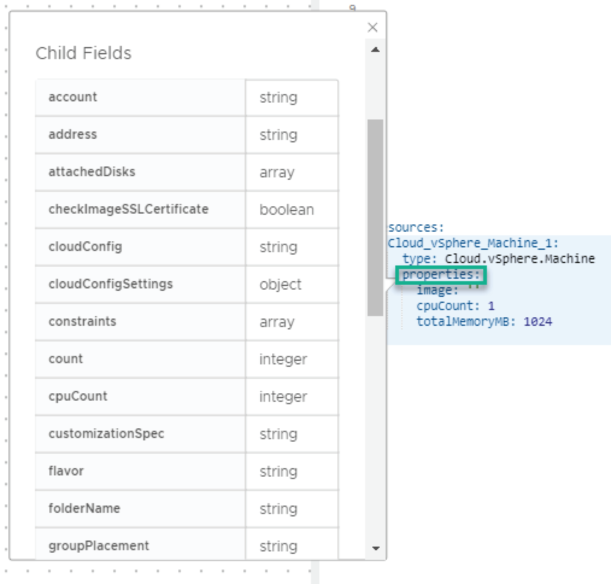


163 Results found for **blueprint.**

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

Introducing the vRA8 YAML Editor

When creating blueprints in past versions of vRealize Automation, 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, I would argue that 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. That being said, unfortunately very little (if any) of what was written supporting vRA7 applies to vRA8 because of the new architecture. There are not very many resources available on the Internet today for helping with a new vRA8 deployment, configuration, and development. What is available often applies to vRealize Automation Cloud (aka vRAC, which is slightly more mature due to its accelerated release cadence) and may or may not apply to the on-prem vRA8 product yet. In another 6 months, when the vRAC features are rolled into the next release of vRA8, those articles will become much more valuable. Until the community has had a chance to build a substantial amount of content surrounding the new platform, this deficit of support content and materials should absolutely be taken into consideration when determining whether or not to deploy the new platform in a production environment.

Tagging, Custom Properties, and Propagation

With vRA8 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, I cannot seem to find them whether in the GUI or the API. I'm not sure of the overall intent behind them, 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

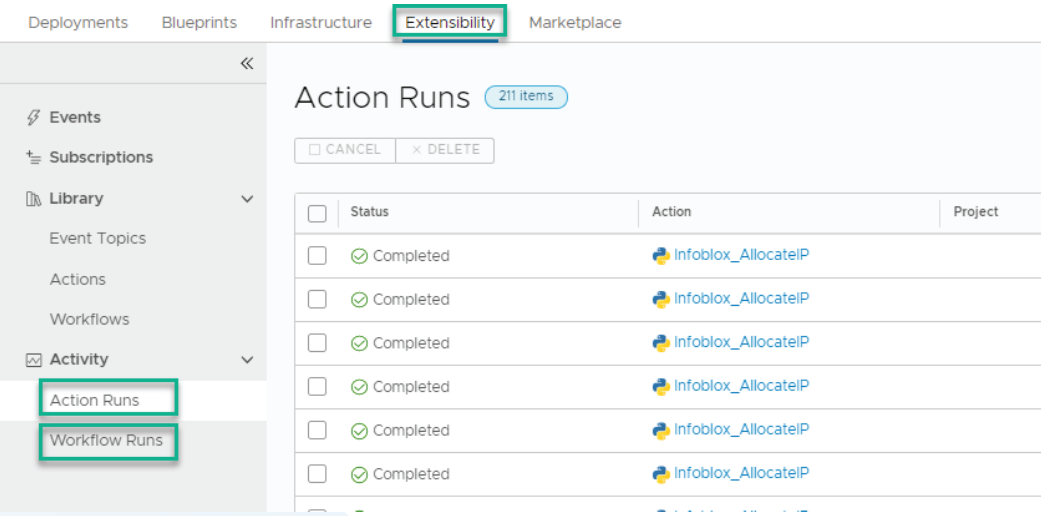
In vRA8, custom properties are not like they were in vRA7. Custom properties can still be applied to a resource to specify metadata about that resource, but they are noticeably missing from infrastructure endpoints, and appear to be limited in total functionality.

From what I can tell, the only tags/custom properties/metadata that propagates down to deployed resources — such as virtual machines — are project resource tags and tags set on the resource itself in the blueprint. I see this change as a major shortcoming in the product. 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.

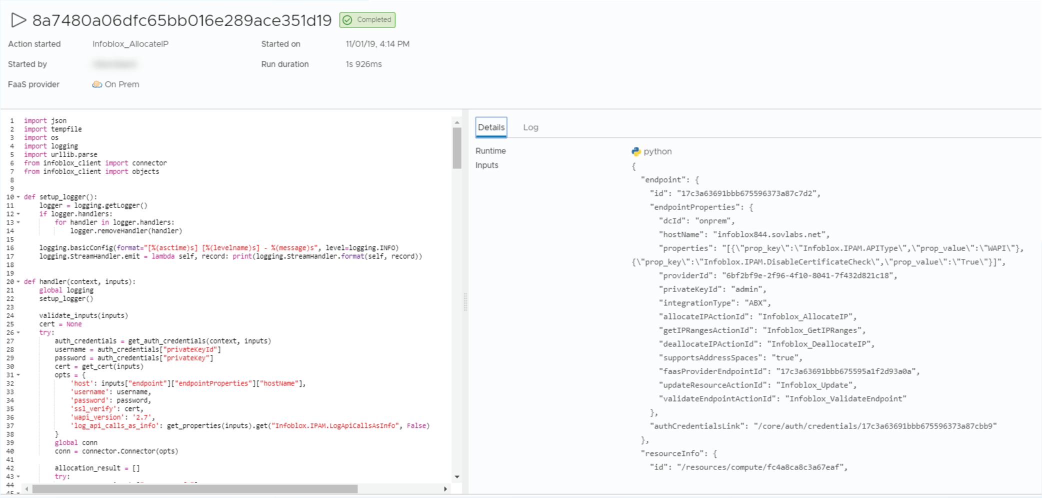
Action Based Extensibility (ABX), vRealize Orchestrator and Extensibility

ABX is the new extensibility offering packaged with vRA8 (in addition to vRealize Orchestrator) that uses a FaaS provider (AWS, Azure and On Prem offered today) to provide extensibility for the new platform. As someone who has spent countless hours working with vRO, ABX is probably one of the more intriguing announcements surrounding vRA8.

One of the things that I am impressed with right out of the gate is that the 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

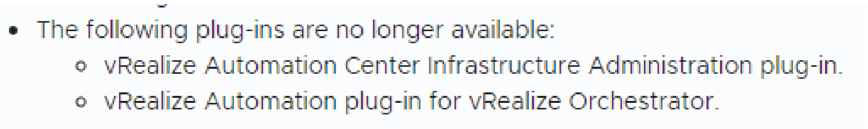
```python
import json
import tempfile
import os
import logging
import urllib.parse
from infoblox_client import connector
from infoblox_client import objects


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-2f96-4f10-8041-7f432d821c18",
      "privateKeyId": "admin",
      "integrationType": "ABX",
      "allocateIPActionId": "Infoblox_AllocateIP",
      "getIPRangesActionId": "Infoblox_GetIPRanges",
      "deallocateIPActionId": "Infoblox_DeallocateIP",
      "supportsAddressSpaces": "true",
      "faasProviderEndpointId": "17c3a63691bbb675595a1f2d93a0a",
      "updateResourceActionId": "Infoblox_Update",
      "validateEndpointActionId": "Infoblox_ValidateEndpoint"
    },
    "authCredentialsLink": "/core/auth/credentials/17c3a63691bbb675596373a87cbb9"
  },
  "resourceInfo": {
    "id": "/resources/compute/fc4a8ca8c3a67eaf",
```

I come from a background heavy in vRealize Orchestrator and am not a true developer. From that point of view, ABX actions and action flows (a way of chaining actions together like a workflow) are certainly learnable; however, without a library of code samples available to start with, it may take a while to learn. In ABX you don't have the standard vCenter or vRA plug-ins that came bundled in recent versions of vRO; in fact, even in vRO8 the vRA plug-in is gone.



- The following plug-ins are no longer available:
  - vRealize Automation Center Infrastructure Administration plug-in.
  - vRealize Automation plug-in for vRealize Orchestrator.

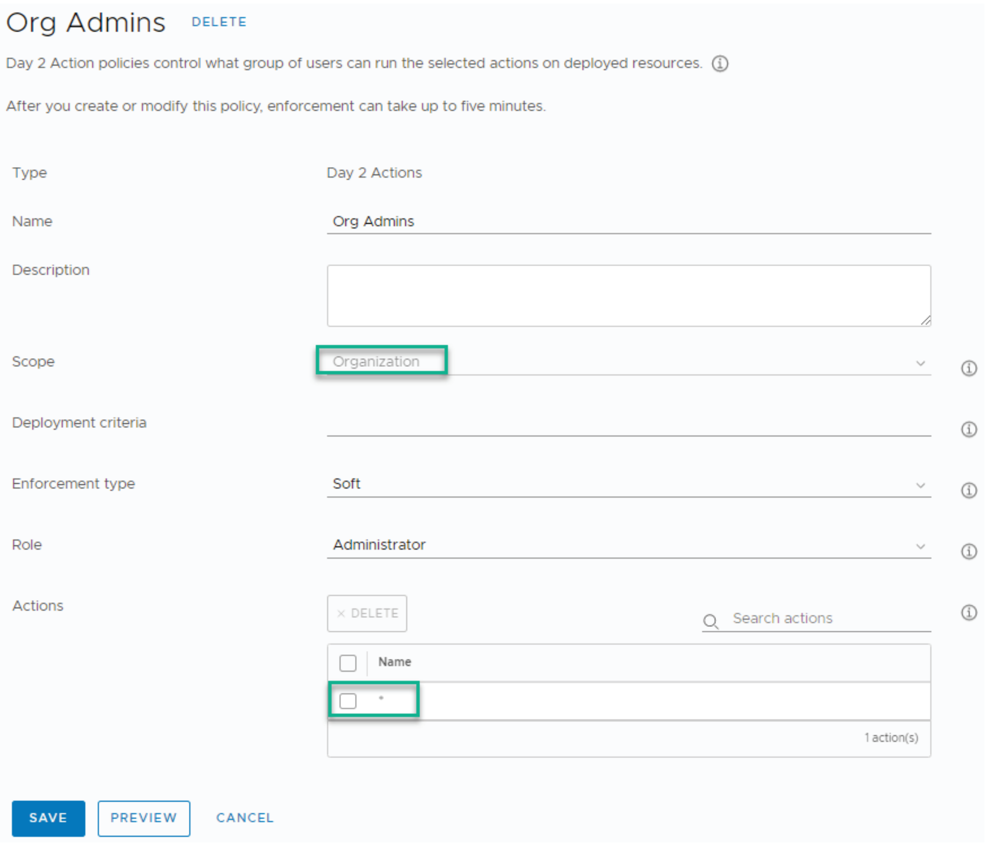
The implications that this change could have on your migrations as a customer could be quite staggering. If you have written a ton of complex extensibility code that centered upon using the vRA plug-in for vRO, you're going to have to re-write all of it using REST. That's not necessarily bad news. Even in vRA7, I personally prefer working with REST over the plug-in, but it will be necessary to re-tool all custom code prior to migrating to the new platform in addition to the potential learning curve for REST if your staff is not already familiar with REST.

vRealize Orchestrator 8 has removed the Java client entirely, and has migrated entirely to a web interface similar to what was available in vRA7.6. While I wasn't a fan of the vRO Java client, I think that the way the workflows were organized in a hierarchical folder structure was very intuitive. In the limited amount of time that I have had to work with the new vRO web client, I have found that trying to find a workflow by search does not always return the results I was expecting. For example, when attempting to "Add a vCenter Server Instance", I couldn't quite recall exactly what the workflow was called. In the old version I could have navigated to Library > vCenter > Configuration to find the exact workflow needed. In the new UI, I ended up opening the old vRO client in 7.6 to find the name of the workflow I needed after 3 failed attempts at searching for the workflow. This scenario very well may be user error, or just old habits that are deeply ingrained from working with vRO since 5.x, but I miss the folders.

Entitlements and Policy Definitions

The entitlement constructs of past versions of vRA 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, I had spent a considerable amount of time writing custom workflows to copy entitlements from a "template" business group out to all of the other business groups (150+). The template business group had three separate entitlements, each one pertaining to a different function (administrative, service account oriented, or end-user). All in all, we had in excess of 450 entitlements and we needed all of them to match the corresponding entitlement tied to the template business group. If we needed to add a single action that we wanted entitled to all 150 of our business groups, we would have to first entitle that action in the template business group, and then run the workflow that copied the entitlements out to the other 150 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 I am able to 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.

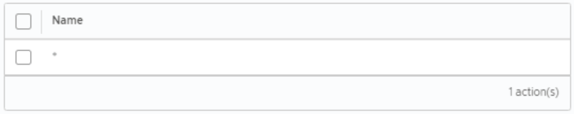
Type	Day 2 Actions
Name	Org Admins
Description	
Scope	Organization
Deployment criteria	
Enforcement type	Soft
Role	Administrator
Actions	× DELETE Search actions

□	Name
□	*
	1 action(s)

SAVE PREVIEW CANCEL

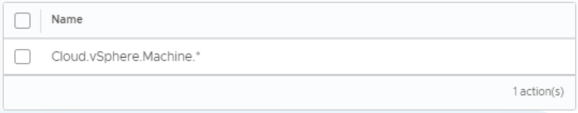
Wildcard Policy Selection: In past versions of vRA7, you didn't have the option to specify, for example, that your administrative group be entitled to all machine level actions. Instead, you had to individually select each of the 20+ actions you wanted to entitle to administrative users of the business 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:



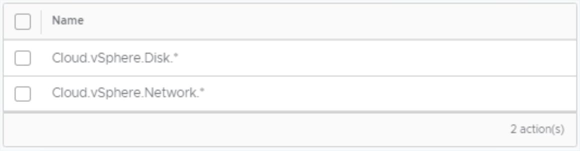
□	Name
□	*
	1 action(s)

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



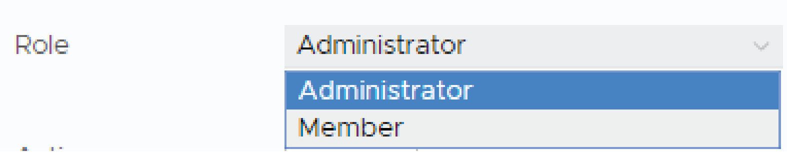
□	Name
□	Cloud.vSphere.Machine.*
	1 action(s)

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)

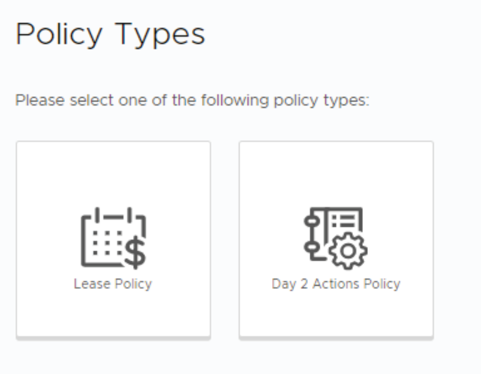
Individual Group or User Policy Assignment: In the previous example with the template business group we had those base level actions, but we also had requirements where we would create specialty entitlements within a business group to entitle specific actions above and beyond the base entitlements that 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 vRA8 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

Administrator
Member

Catalog Item Policies: When you go to create a new policy, you may notice that the capability to restrict access to Catalog Items in vRA8 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:

Lease Policy

Day 2 Actions Policy

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.

vRealize Automation 8 is here. 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 vRealize Automation world is on version 8, or you've worked with vRA/vCAC since the DynamicOps days, there is a learning curve ahead. Start to work with vRA8 now in development and stay aware of updated releases as you methodically approach production deployment and determine whether or not the platform is mature enough to address all of your use cases today, or if you require additional functionality coming in future versions.

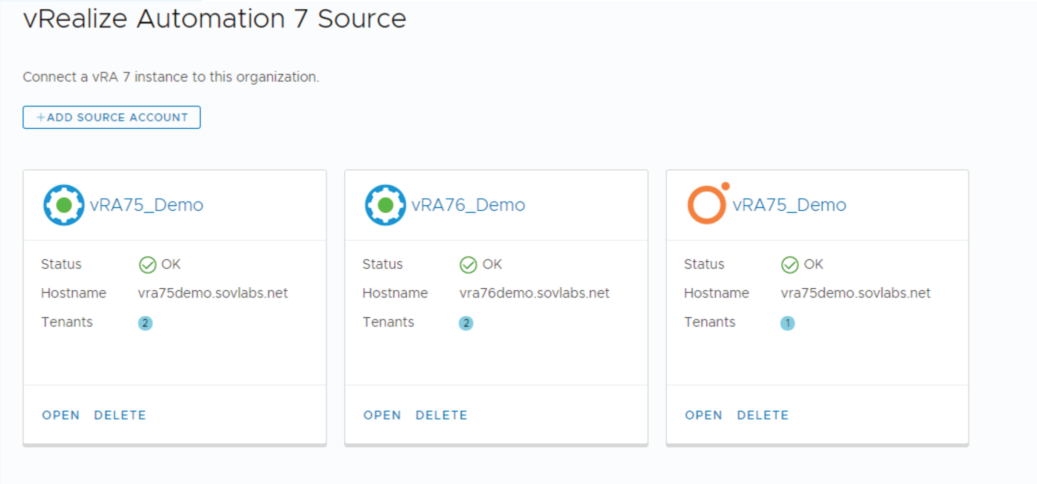
An Overview of the vRA8 Migration Assessment Tool

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 your first major vRA migration. In an ideal world these migrations would only take a few clicks of the mouse. Migrating from vRA7.x to vRA8.x 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 have to identify everything that will require hand-holding 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. As of this writing, we have not yet learned if the migration assessment tool will determine if your customizations and workloads are ready for migration.

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. As of this writing there is no option to validate against a non-embedded vRO, which most folks are running in production these days. For my testing, I configured a vRA7.5 and a vRA7.6 environment to see how they look for migration. I 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

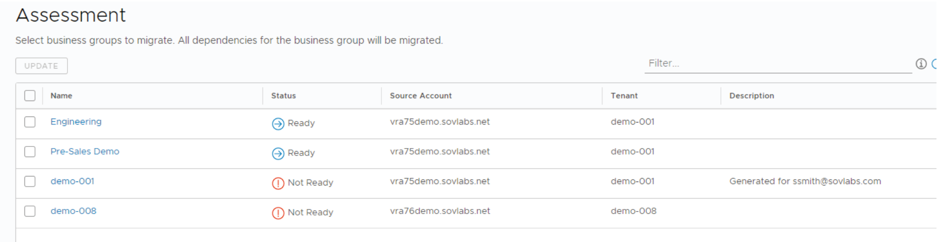
vRA76_Demo

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 my case (below) there are 4 business groups, but only 2 of them have any blueprints. Can you guess which two they are?



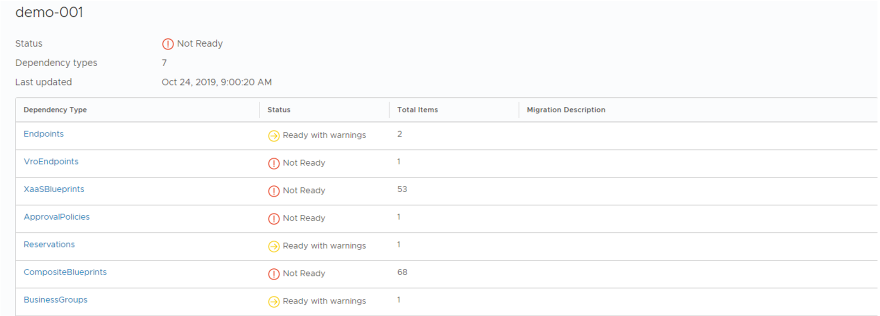
Assessment

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

☐	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; however, this information leads me to believe that this migration assessment tool will be the tool used in the future to perform such tasks.

Once you open up 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.



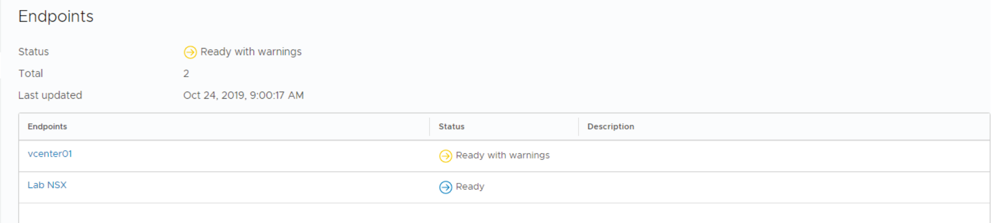
demo-001

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 my environment. I have two Endpoints and one is Ready and one is Ready with Warnings.

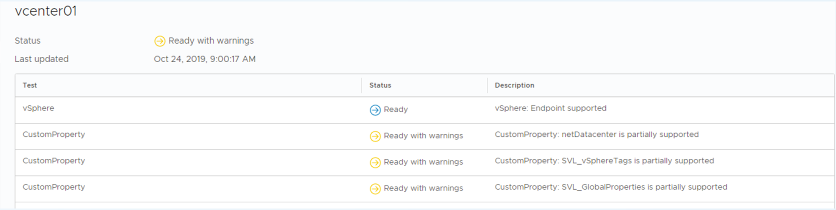


Endpoints

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 further investigation into the endpoint with warnings, I see that the warnings are all related to custom properties defined on the EndPoint. This explanation makes sense considering properties are almost non-existent in vRA8, and you cannot define properties on vRA8 Endpoints. What I find to be a little misleading is the migration assessment tool shows that the properties are "partially supported." I searched the documentation to distill what that means, but I could not find any explanation. My best guess is that you can define tags within vRA8 and use tags in place of properties, which makes them "partially supported."



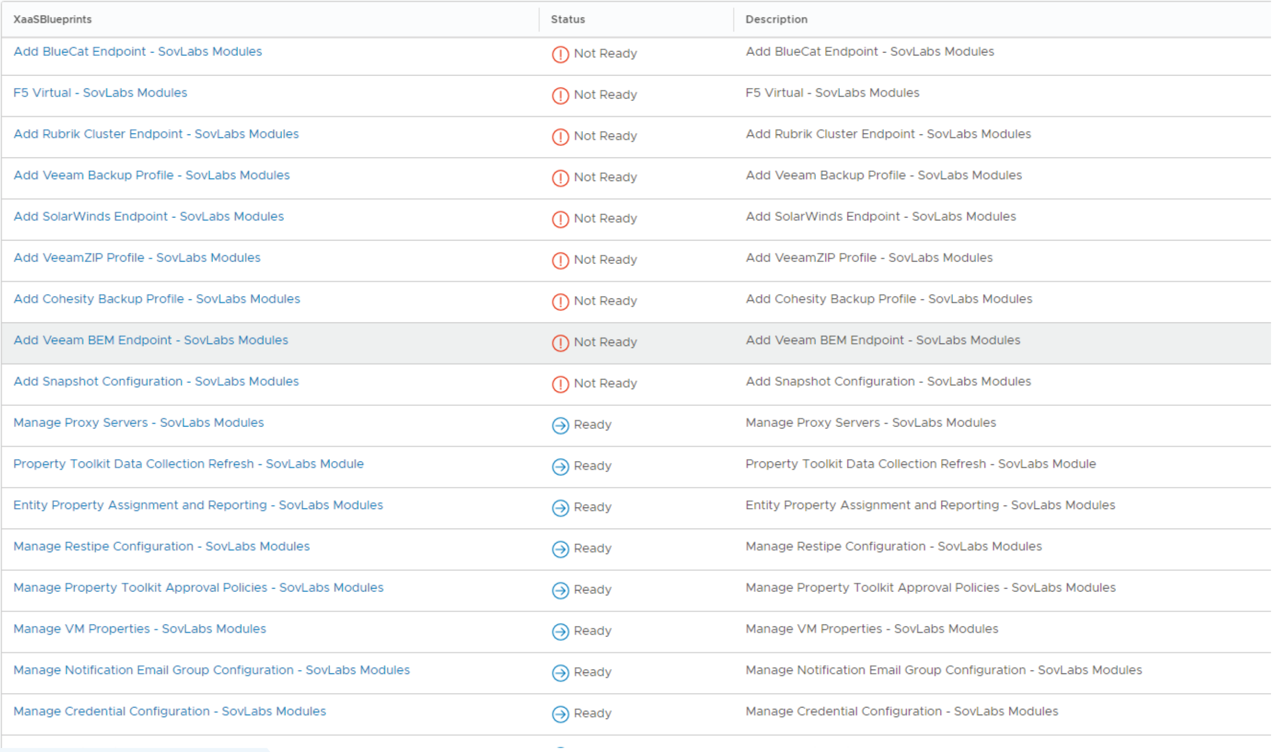
vcenter01

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 I investigate items under XaaS Blueprints (see image below) I see a large number of SovLabs 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 or not the blueprint uses vRO custom resources. In my opinion, that criteria is not a great way of determining and reporting if something is ready to migrate to vRA8. Furthermore, I reviewed the list of XaaS Blueprints that it says are ready and one by one I realized they would not work with vRA8.



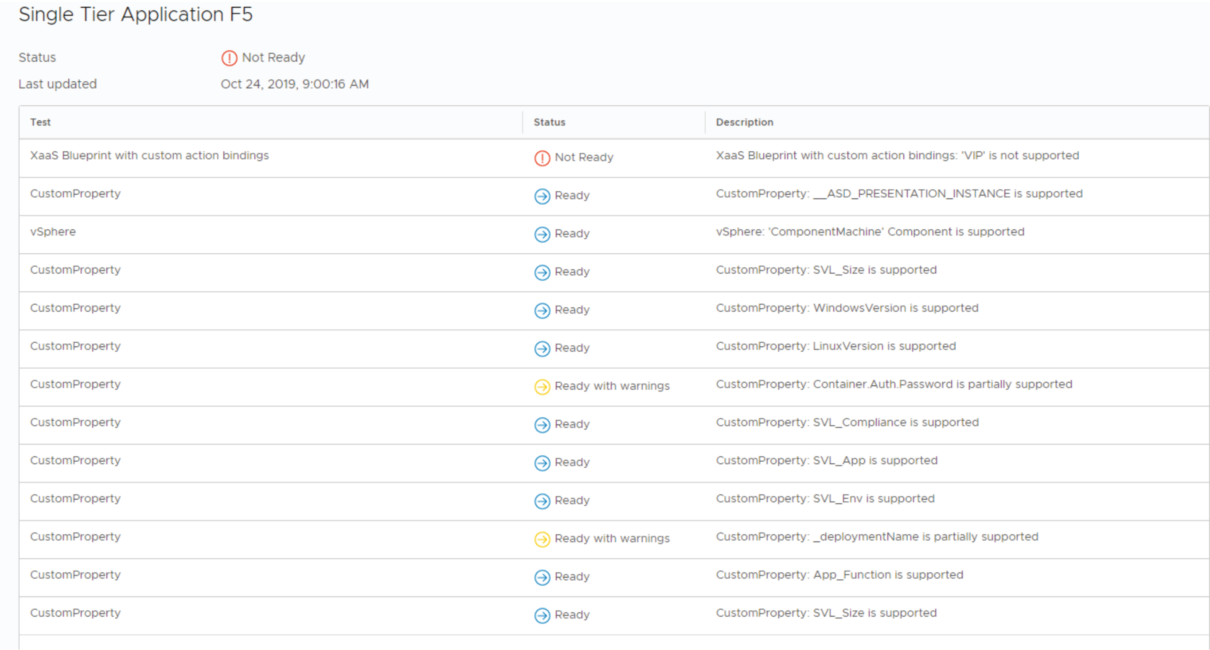
XaaSBlueprints	Status	Description
Add BlueCat Endpoint - SovLabs Modules	Not Ready	Add BlueCat Endpoint - SovLabs Modules
F5 Virtual - SovLabs Modules	Not Ready	F5 Virtual - SovLabs Modules
Add Rubrik Cluster Endpoint - SovLabs Modules	Not Ready	Add Rubrik Cluster Endpoint - SovLabs Modules
Add Veeam 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 vRA7 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

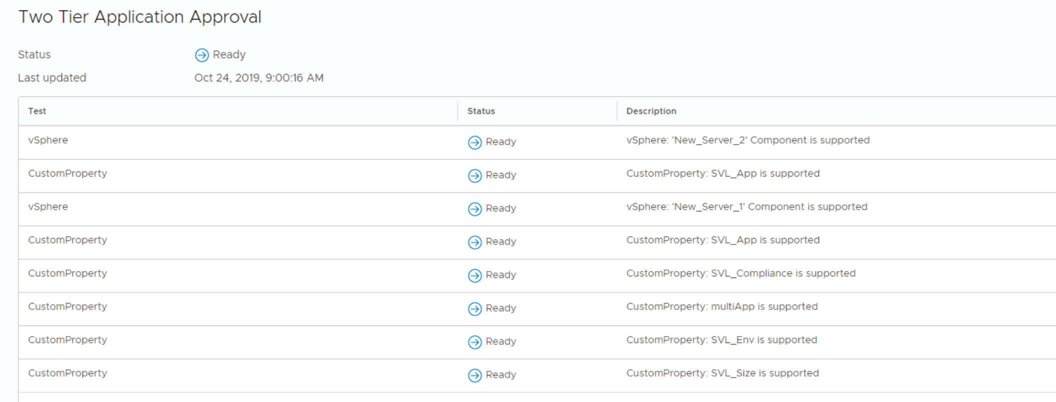
Test	Status	Description
XaaS Blueprint with custom action bindings	ⓘ Not Ready	XaaS Blueprint with custom action bindings: 'VIP' is not supported
CustomProperty	→ Ready	CustomProperty: __ASD_PRESENTATION_INSTANCE is supported
vSphere	→ Ready	vSphere: 'ComponentMachine' Component is supported
CustomProperty	→ 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. All SovLabs properties will be marked as ready and all properties you may have created on your own will be marked as ready. You will have to do your own due diligence to determine what these properties are used for.

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 vRA7, there are a number of 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. vRA8 is a completely new platform, written from the ground up, and no longer includes the IaaS host that controlled all of these states in vRA7.

vRA8 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.

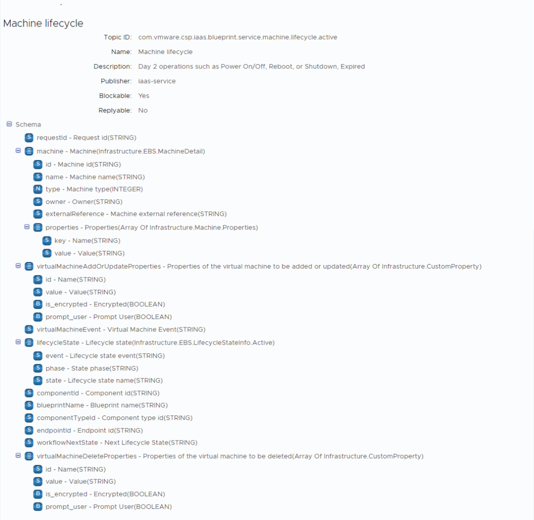
- Deployment Level
    - Deployment Requested
    - Compute Reservation
    - Compute Allocation
    - Network Configure
    - Deployment Completed
- Network Level
    - Deployment Resource Requested
    - Deployment Resource Completed
- Machine Level
    - Deployment Resource Requested
    - Compute 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. For more information on the new event states in vRA8 check out vRealize Automation 8 Extensibility Event Topic Order of Operations by Mike Bombard.

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

Publisher: iaas-service

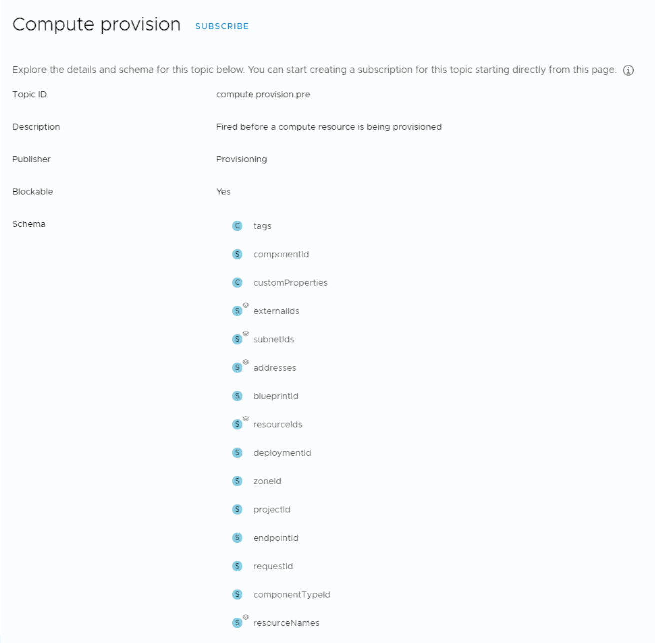
Blockable: Yes

Replyable: No

```
⊟ Schema
    S requestId - Request id(STRING)
  ⊟ ☰ machine - Machine(Infrastructure.EBS.MachineDetail)
      S id - Machine id(STRING)
      S name - Machine name(STRING)
      N type - Machine type(INTEGER)
      S owner - Owner(STRING)
      S externalReference - Machine external reference(STRING)
    ⊟ ☰ properties - Properties(Array Of Infrastructure.Machine.Properties)
        S key - Name(STRING)
        S value - Value(STRING)
  ⊟ ☰ virtualMachineAddOrUpdateProperties - Properties of the virtual machine to be added or updated(Array Of Infrastructure.CustomProperty)
      S id - Name(STRING)
      S value - Value(STRING)
      B is_encrypted - Encrypted(BOOLEAN)
      B prompt_user - Prompt User(BOOLEAN)
    S virtualMachineEvent - Virtual Machine Event(STRING)
  ⊟ ☰ lifecycleState - Lifecycle state(Infrastructure.EBS.LifecycleStateInfo.Active)
      S event - Lifecycle state event(STRING)
      S phase - State phase(STRING)
      S state - Lifecycle state name(STRING)
    S componentId - Component id(STRING)
    S blueprintName - Blueprint name(STRING)
    S componentTypeId - Component type id(STRING)
    S endpointId - Endpoint id(STRING)
    S workflowNextState - Next Lifecycle State(STRING)
  ⊟ ☰ virtualMachineDeleteProperties - Properties of the virtual machine to be deleted(Array Of Infrastructure.CustomProperty)
      S id - Name(STRING)
      S value - Value(STRING)
      B is_encrypted - Encrypted(BOOLEAN)
      B 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 are building vRO workflows for vRA, 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 vRA plugin for vRO also has a number of prebuilt actions and workflows that you can use to make your life easier. *Easier* if your workflows are built for vRA7.x because no plugin exists in vRA8 today. If VMware were to ever come out with a new plugin for vRA8, it would be vastly different than the current plugin and wouldn't help much in the way of migration.

Today, if you want to interact with vRA8, you will need to use the vRO Rest plugin and do so through the Rest API.

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 vRA7 would be somewhat transferable to vRA8. 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. If you answered yes to more than one of the questions, then you will need to start from the ground up.

Or will you?!?

There is another option. I have been helping many vRA customers migrate away from their customizations on vRA7 to the modules produced by SovLabs in preparation for migrating to vRA8. Whether you are solely using lifecycle workflows, or using XaaS Blueprints in vRA to meet your businesses needs, we can help. Migrating your customizations to policy-driven software (Yes, SovLabs software... shameless plug) now will allow you to avoid having to rewrite code. And, migrating your SovLabs configurations from vRA7 to vRA8 is seamless.

Whatever you choose, don't wait until the last minute to start planning.

Identifying Important Gaps in vRA 8 Features, Functionalities, and Integrations

As I began to draft this content, I took some time to reflect on December of 2015. At the time I was involved with systems architecture for a large organization where we had rolled out vRA 6.2 in production for the first time about 5 months prior. vRealize Automation 7 had just been announced, and we were ready to start working with the new platform to put it through its paces. We installed vRA 7.0 within the next month or two and pretty early on in the process identified that the product was not capable of synchronizing our existing security groups with the solution because of the organization's policy of having a special character included in the name of the majority of our security groups. We were very familiar with this issue because, when we first started working with an earlier version of vRA 6.2, we had run into the same issue. That issue had been remedied in a later version of 6.2, only to have the same issue resurface again in 7.0. It wasn't until 7.2 that the special character issue was resolved, and then it took us another *8 months* to restructure our blueprints, re-write custom code, and migrate the entire environment.

I tell this story because I believe that it is imperative that you start working with vRA8 no later than version 8.1 in a development environment. With the vRA8 product being redesigned from the ground up, you will want to identify potential deficiencies early on so that they can be addressed in the product. The purpose of this article is to highlight some of the high level features, functionalities, and integrations that we have already encountered that are noticeably missing in vRA 8.0, so that you can focus your efforts on other areas that may be important to your organization.

Before getting started, I want to note that some of these gaps have already been acknowledged by VMware, and are (hopefully) being addressed in vRA 8.1 or 8.2. Our primary goal with this article is to be sure all of the major gaps we have identified in the product are already clearly defined, so that vRA users are well educated when making the decision whether or not the time is right for them to migrate to the new platform.

Approval Policies

If you are using approval policies today in vRA7, or if you require that requests are appropriately governed and are considering vRA8 for your enterprise deployment, you need to be aware that approval policies do not exist in vRA8. Many of the customers we work with rely on the approval policy constructs built into past versions of vRA. Again, approval policies do not exist in vRA8.

Multi-Tenancy

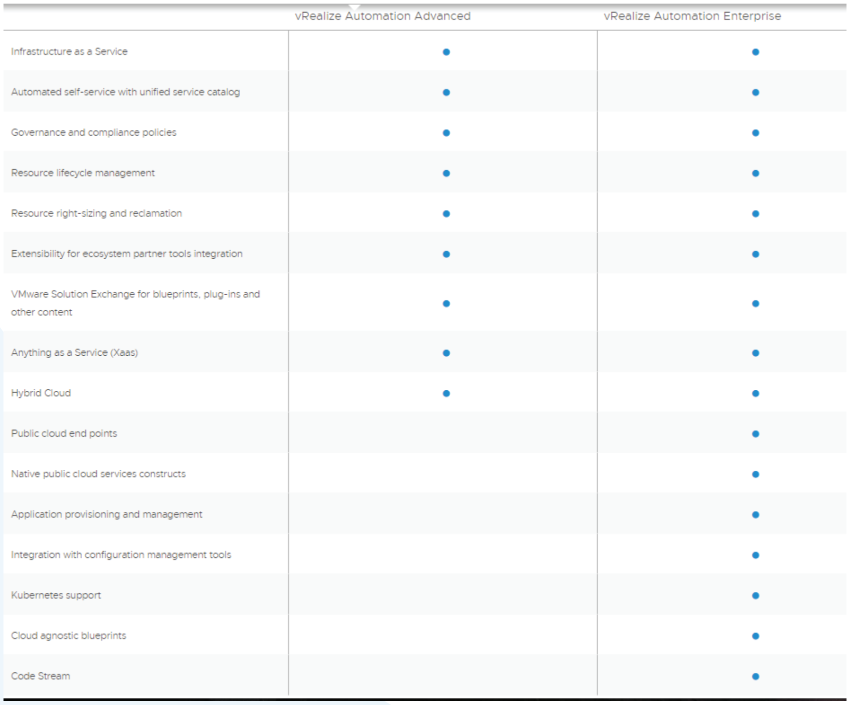
In vRA 8.0, multi-tenancy is not available. VMware has hinted that multi-tenancy should be reintroduced in vRA 8.1. In vRA 8.0 you can even see that the constructs exist for something called organizations. Because it will still be a few months until anyone is able to see how VMware will approach the idea of multi-tenancy in vRA 8, there is still quite a bit of investigation required once multi-tenancy is released.

Property Dictionary

If you were leveraging the property dictionary, property groups, and property definitions in vRA7, you need to know that there is no property dictionary — or its equivalent — in vRA 8.0. In addition, custom properties are no longer available on most of the endpoints in vRA8 (think Endpoints, Compute Resources, etc.) for propagation down to provisioned resources (see this post for more detail).

Licensing Differences

VMware has added many new capabilities with the new platform. You should know that most of the big-hitter type capabilities — including AWS, Azure and Google Cloud deployments, Kubernetes support, and Code-Stream integration — all require a vRA8 Enterprise license. If your organization requires basic IaaS and XaaS type deployments to on-prem resources only, you will only need Advanced-level licensing. However, most of the newly added features that have been focused on in the marketing campaigns to date will require vRA Enterprise licensing. Here is the link to the official vRA page showing the difference in the two versions. I have included a screenshot here as well.



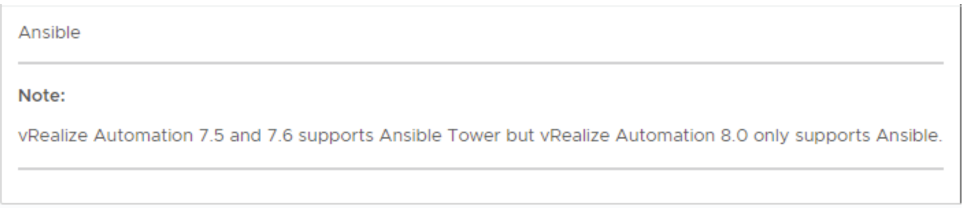
	vRealize Automation Advanced	vRealize Automation Enterprise
Infrastructure as a Service	●	●
Automated self-service with unified service catalog	●	●
Governance and compliance policies	●	●
Resource lifecycle management	●	●
Resource right-sizing and reclamation	●	●
Extensibility for ecosystem partner tools integration	●	●
VMware Solution Exchange for blueprints, plug-ins and other content	●	●
Anything as a Service (Xaas)	●	●
Hybrid Cloud	●	●
Public cloud end points		●
Native public cloud services constructs		●
Application provisioning and management		●
Integration with configuration management tools		●
Kubernetes support		●
Cloud agnostic blueprints		●
Code Stream		●

Configuration Management

Much of the marketing for vRA8 points to the fact that your configuration management tools will continue to be available directly from the blueprint design canvas. You can drag and drop configuration management blueprint components directly on to your canvas when designing a blueprint. As in vRA7 though, you should note that this feature is only available in the Enterprise version of vRA8, and that, depending on the CM tools that you are using the integrations, is quite a bit different than what was available in vRA7.

Ansible Integration

The first thing to note with the vRA8 Ansible integration is that vRA8 supports only Ansible Open Source, not Ansible Tower.



Ansible

Note:

vRealize Automation 7.5 and 7.6 supports Ansible Tower but vRealize Automation 8.0 only supports Ansible.

Opting to integrate Ansible in this fashion does not allow execution of day two operations from Ansible, role-based access control, the capability to schedule jobs, graphical inventory management, or external logging integrations. All of these functionalities are possible if you are using Ansible Tower.

Puppet

The Puppet integration will allow you to specify roles, hosts, and credentials, but it does not offer the ability to invoke the Puppet agent as many times as you'd like. The Puppet Integration also does not allow you to use alternative node classification, drive hiera, factor, specify alternative exit codes, or parse the logs for regular expressions that would signify a success without requiring an exit code. The native Puppet integration in vRA8 is rudimentary and perfunctory, but not at all robust or tolerant of real world conditions in which modules may not perform as expected 100% of the time.

There is a ton of added functionality included in vRA8. With the addition of Cloud Agnostic blueprints, support for Kubernetes, and Code-Stream capabilities built in, there is a lot to look forward to in this new offering. However, there are also several feature gaps in vRA 8.0. Some organizations are ready to use the product today, while most organizations may be better off waiting until some of these feature gaps are mitigated.

Limited XaaS Functionality in vRA 8

I want to start off by mentioning that VMware is listing Anything as a Service (XaaS) as being "built in" to vRA 8.0. The current implementation of XaaS in vRA8 is only a subset of the functionality that is available in vRA7. In the second part of this series, I am going to point out why the current implementation of XaaS in vRA8 falls short of where it should be before any customer requiring XaaS considers using it in a production environment.

I will also note that we see a lot of customers using XaaS in vRA7 to customize the platform so that it meets their organization's specific requirements. To level-set for this discussion, here are the major features/capabilities available in vRA7:

1. XaaS Blueprints that do not create a catalog item
2. XaaS Blueprints that do create manageable Catalog Items
3. XaaS Day 2 Actions (Resource Actions)
4. Custom Resources - ability to create anything as a provisioned manageable object in vRA
5. Resource Mappings - ability to correlate between vRA Catalog Resource type and a vRO

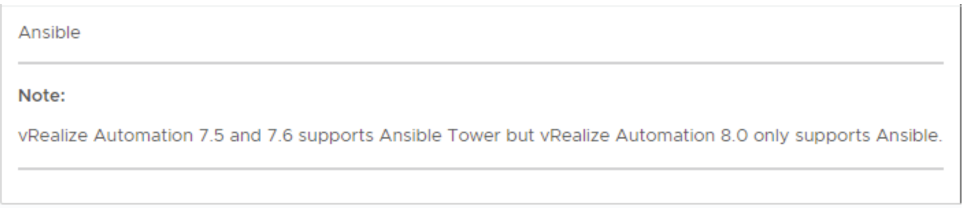
type for custom resource actions

6. Component Lifecycle - ability to define lifecycle (Provisioning, Update, Destroy) actions for an XaaS Blueprint with a Custom Resource output type.
7. Ability to call XaaS blueprints inside of other blueprints (Nested XaaS Blueprints)

Today, the only feature of XaaS that is available in vRA 8.0 is item #1 above. The bulk of what XaaS does in vRA7 is not currently available in vRA 8.0. Also note that, as I speak about XaaS, I am referring to the product specifically as it pertains to vRO, not the ABX component. From what I have been able to uncover with ABX, it certainly doesn't appear ready to support any of the advanced XaaS capabilities above, although it does appear that you can address item 1 in the list above using ABX today.

What is available in vRA 8 XaaS?

Before we talk about what is missing, I want to highlight what you get with XaaS in vRA 8.0 by way of a simple workflow example. The way that XaaS is implemented in vRA 8.0 today, it is configured from Service Broker (not from Cloud Assembly; more discussion on that later). If you navigate to *Content & Policies* > *Content Sources*, you can add a *New Content Source* that matches one of the following:

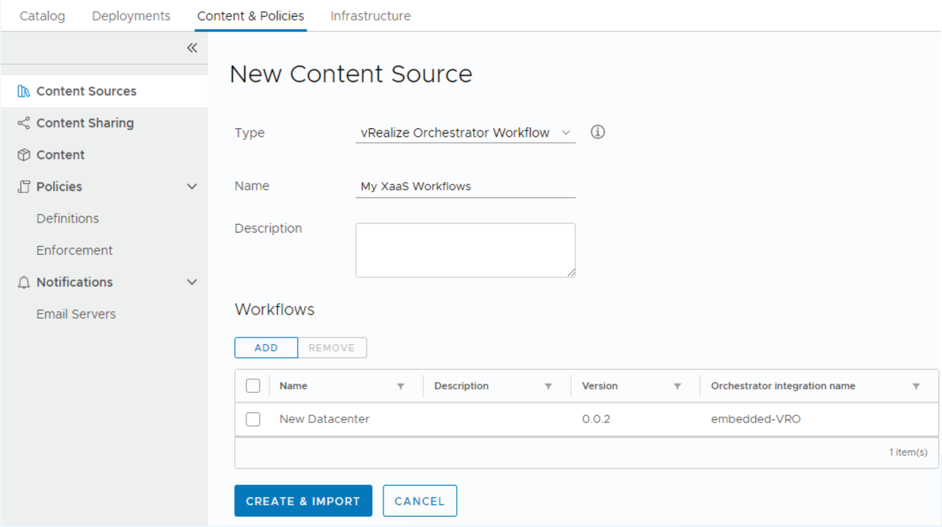


Ansible

Note:

vRealize Automation 7.5 and 7.6 supports Ansible Tower but vRealize Automation 8.0 only supports Ansible.

Since we are working with XaaS (specifically with vRO) we select the vRealize Orchestrator Workflow option. I can then select the workflows I want to expose with this content source (Note that only one workflow can be exposed via a single content source). I select a workflow called New Datacenter, and then "Create & Import" my content source.



New Content Source

Type: vRealize Orchestrator Workflow

Name: My XaaS Workflows

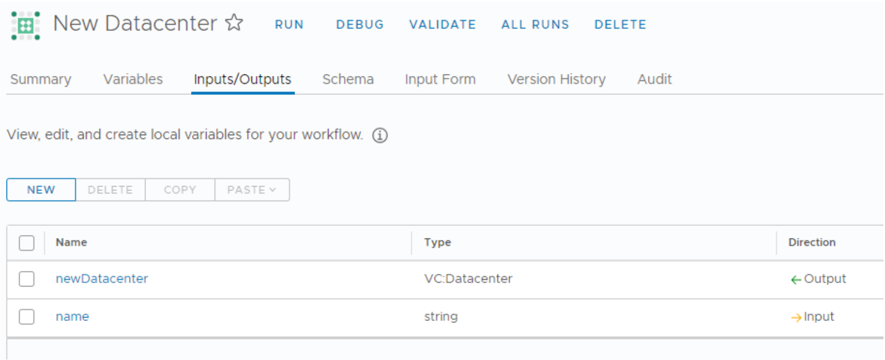
Description:

Workflows

□	Name	Description	Version	Orchestrator integration name
□	New Datacenter		0.0.2	embedded-VRO

1 item(s)

The New Datacenter workflow is a very simple workflow, in which the user submits the name of a new datacenter they would like to create in vRO. The outputs of the workflow include the actual VC:Datacenter object that was created in my vCenter.



New Datacenter ☆ RUN DEBUG VALIDATE ALL RUNS DELETE

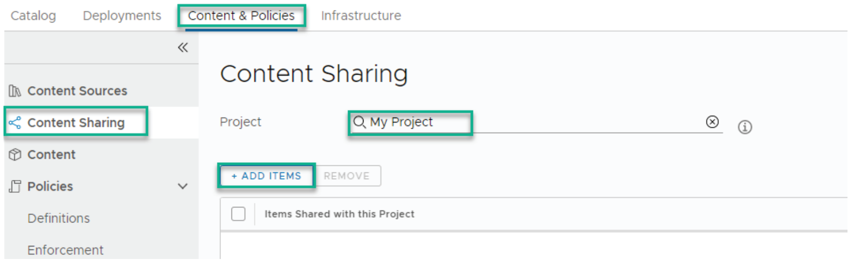
Summary Variables Inputs/Outputs Schema Input Form Version History Audit

View, edit, and create local variables for your workflow.

NEW DELETE COPY PASTE

□	Name	Type	Direction
□	newDatacenter	VC:Datacenter	← Output
□	name	string	→ Input

To share the content with your project, you go to *Content Sharing*, and then *Add Items*.



Catalog Deployments Content & Policies Infrastructure

Content Sources
Content Sharing
Content
Policies
Definitions
Enforcement

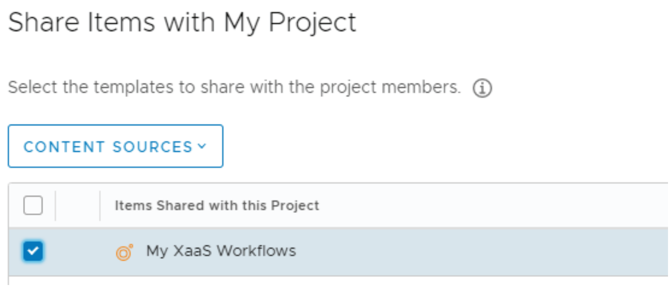
Content Sharing

Project: My Project

+ ADD ITEMS REMOVE

□ Items Shared with this Project

From here, I then select the My XaaS Workflows Content Source I created earlier.



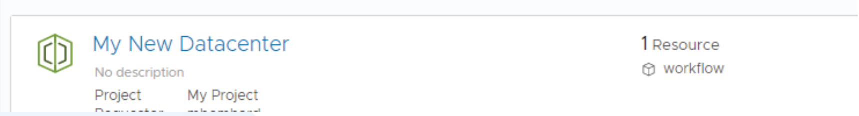
Share Items with My Project

Select the templates to share with the project members.

CONTENT SOURCES

□	Items Shared with this Project
☑	My XaaS Workflows

At this point I can access the New Datacenter Request from my catalog, I can submit a request, and I can see that the request completes successfully.



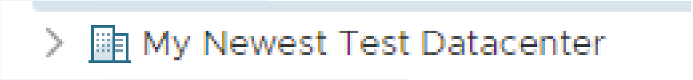
My New Datacenter

No description

Project: My Project

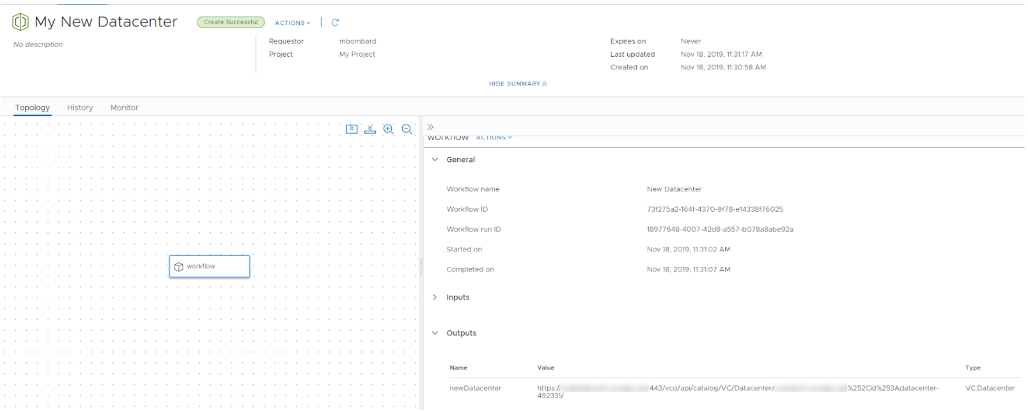
1 Resource
workflow

I can also log in to vCenter and view the Datacenter that I requested.



My Newest Test Datacenter

That is what I have been able to identify as the extent of what XaaS in vRA 8.0 is capable of today. You can import a vRO workflow into Service Broker, expose that workflow in the catalog, and allow the user to submit the workflow from vRA. That workflow can even be used to create things (a datacenter in vCenter in this example), but XaaS is currently incapable of returning that object back to vRA to become a managed object in vRA itself. You'll note below that the resource that shows on the canvas for this deployment is a "workflow", rather than the actual datacenter object that we created. You can see the inputs and outputs for the workflow, but there is nothing of value added to our catalog. That datacenter is not added into the vRA lifecycle, nor can it have actions executed against it.



My New Datacenter

No description

Requestor mbombard
Project My Project

Expires on Never
Last updated Nov 18, 2019, 11:31:17 AM
Created on Nov 18, 2019, 11:30:58 AM

General

Workflow name New Datacenter
Workflow ID 73f275a2-164f-4370-9f78-e14338f76025
Workflow run ID 18977648-4007-42d8-a557-b078a8abe92a
Started on Nov 18, 2019, 11:31:02 AM
Completed on Nov 18, 2019, 11:31:07 AM

Inputs

Outputs

Name	Value	Type
newDatacenter	https://[redacted]:443/vco/api/catalog/VC/Datacenter/[redacted]%252Cid%253Adatacenter-492331/	VC:Datacenter

What is not available in vRA 8 XaaS?

Custom Resources

I'm going to tackle this subject first, because the lack of availability of this feature very likely plays into why the other features are also missing. Without the capability to allow vRA to recognize vRO types, there is no way that vRA can understand what the items are that are being provisioned. Also, vRA would not be able to create a manageable catalog item, allow you to call a day 2 action on that item, or define a lifecycle around that item. In vRA 8.0, Custom Resources are not currently available.

As an example, if your organization has a requirement to automate the creation of a new user account, you could use the vRO Active Directory plugin to create a new user account. The workflow may have an output type of *AD:User*, in which the newly created user account can be passed out of the workflow. Where custom resources come into play here is that the output type of *AD:User* is not something that the vRA catalog understands right out of the box. A custom resource allows you to make vRA aware of the vRO "type" so that an *AD:User* can be created as a manageable vRA catalog resource. You can place leases around that object, you can govern any second day actions presented to that object, you can create and present actions on that object to allow the *AD:User* to be deleted directly from the vRA portal. You are able to treat that *AD:User* just like any other catalog resource by automating the entire lifecycle of that *AD:User* object.

Dynamic Types

Where Custom Resources become even more powerful is when you take into consideration the dynamic types plugin available in vRO. The dynamic types plugin allows you to create new types that are defined by the user without having to understand the vRO SDK; they are defined through a series of vRO workflows and actions written in vRO itself. Side note: see earlier article covering dynamic types here. With the dynamic types plugin, you could define a type for anything you can dream up, and then wrap that object in the lifecycle control processes available in vRA. If your endpoint vendor does not have a vRO plugin available for the management of their resources, but they do have a REST API available, you could create a Dynamic Type for the resource created by a REST call to that endpoint so that the resource is now visible in, managed by, and included in the vRA lifecycle.

Third-Party Vendor Custom Types

In addition, without the availability of Custom Resources, third-party vendors are restricted in their ability to present their content directly in the vRealize Automation Catalog. Also, a third-party vRO plugin that has a Custom Type defined in vRO cannot be pulled into the vRA catalog by a user.

Day 2 XaaS Actions

The capability to create custom vRO workflows and then call those workflows directly from a resource in the catalog as a second day action does not exist in vRA 8.0. We have seen quite a few customers that identify this specific functionality as a crucial requirement in their decision making process on when to migrate to vRA8. The out of the box vRA actions do not meet the requirements for their organization (or they have custom actions to perform for their deployments), whereas the XaaS Second Day Actions in vRA7 met this requirement.

Resource Mappings

The ability to correlate between a vRA Catalog Resource type and a vRO type for custom resource actions does not exist in the present state of the product.

Nested Blueprints

In addition to XaaS Blueprints being inaccessible in Cloud Assembly, even if a developer were to create an XaaS blueprint in Service Broker, they cannot consume that XaaS blueprint as an additional canvas item because nested blueprints are not currently available in the product.

XaaS is Only Available from Service Broker

As I understand it, one of the largest drivers behind the development of vRA8 using industry accepted standards for YAML blueprint development was to enhance the vRA experience as it pertains to organizations with a DevOps-heavy focus. vRA 8 has made massive strides in making the Cloud Assembly portion of vRA8 very accessible for the DevOps-minded organization. Your developers now have the access and capabilities to do some really cutting edge blueprint development and to be able to test and deploy resources directly from Cloud Assembly without ever having to work with a service catalog (Service Broker). Having seen these forward-focused improvements, I was somewhat surprised to learn that the ability to utilize XaaS blueprints is only available in Service Broker, and not from Cloud Assembly itself.

XaaS Summary

If your organization's requirements for XaaS falls into one of the buckets listed above, I would certainly proceed with caution when considering vRA 8.0 for production deployments. Instead, I recommend keeping an eye out for improvements in the XaaS functionality in vRA 8.1 and 8.2.

Unraveling Your Customizations in vRA7 to Prepare for vRA8

Although you may not be ready to make the move to vRealize Automation 8 just yet, it's time to start planning. Time to start unraveling all the customizations and integrations that make your vRealize Automation 7 environment perform it's magic. If you're like most vRA7 users, it is a combination of third party integrations, open source customizations, home grown customizations, and scripts joined together around your specific business process that makes the automation magic happen.

Many vRA7 environments are grown organically over time, with new integrations and customizations added as the needs arise, and little — if any — documentation on exactly what they are and how they work. So where do you begin? My recommendation is to start at the beginning.

Blueprints

You will first need to take inventory of your blueprints. Documenting which ones overlap and which have differences. Some may have different custom forms with unique logic embedded within. Inputs may vary drastically from one to the next. Understand what those differences are and why they are there. What happens to those inputs after they are submitted? It's important to break these inputs down into groups so you can save time by understanding which blueprints have shared integrations and which ones have unique integrations.

System Custom Properties

Many of you are very familiar with the vRA Custom Properties Reference. vRA8 does not currently have an equivalent resource that guides you through how to properly use custom properties in the new product, and, from all accounts, it appears that the custom properties used in vRA7 are no longer valid. If you are using any of the VMware defined system custom properties in your environment, you will need to make a determination in the new platform on how to address each of the transforms. Each one of these properties is setting some type of customization for your provisioned resource, and the mechanism for applying similar customizations in the new product is different.

Custom Forms

Although Custom Forms have only been available since vRA 7.4, they were quickly adopted because custom forms allowed many users to solve challenges they were previously unable to address. Your custom forms may have unique business logic included within them. It's important not to overlook this logic because it may play a very important part in your overall automation outcome. Determine which blueprints have custom forms and thoroughly review each one, documenting any embedded custom logic and what its purpose is.

XaaS Blueprints

Many vRA7 users have adopted the use of XaaS Blueprints before the release of custom forms to allow for a more customized request interface and to accommodate custom business logic into their vRA7 solutions. These XaaS Blueprints can include business logic, integrations that need to run pre-request, and logic to drive outcomes such as which composite blueprint is ultimately requested. XaaS Blueprints can be the most involved part of your assessment.

I've seen vastly different adoption of these blueprints from a single XaaS Blueprint that is used as the sole mechanism for making vRA requests to environments with XaaS Blueprints for every different Composite Blueprint in the environment. These XaaS Blueprints often contain custom vRO actions, which makes them even more complex. XaaS Support in vRA8 is very limited, with no support for Resource Mappings, Custom Resources, Dynamic Types, and other features you may be using in vRA7. To migrate to vRA8, you should plan to eliminate your XaaS Blueprints altogether. For more information on what XaaS support in vRealize Automation 8 check out Identifying Important Gaps in Features, Functionality, and Integrations Part 2 of 2.

Day 2 XaaS

You probably created XaaS items to be used as Day 2 actions. This capability currently doesn't exist in vRA8, so it's critical to review and understand the importance of these capabilities to your organization. Review all of your Day 2 XaaS functions, and make a note of all the supporting vRO workflows that they utilize.

Event Broker Subscriptions

Next, you will want to review your event broker subscriptions. Don't look on the surface. These subscriptions may have rules that are used to determine when these events are executed. This understanding is important because not all subscriptions may be used for every deployment. You will need to understand what rules are in place to be able to map these subscriptions to their actual function. As part of this review, you will also need to figure out what workflow is being called as part of the subscription.

vRO Workflows

At this point, things get more complicated. By now you should have a list of all your vRO workflows that will need to be reviewed. As we have discussed in A first look at vRA8 Migration Assessment Tool Part 2 of 2, there are a number of key changes that will require you to significantly rework these workflows or perhaps even recreate them from scratch. Therefore, it's important to be very thorough when doing this review.

You need to understand what properties are being used in each of your workflows, because these properties are used to drive the behavior of the workflow. In vRA8, not all properties are available at every state. This change is very critical. If the workflow that you are reviewing is executed at a specific state, and needs access to specific information, this information may not be available at the point in time in vRealize Automation 8.

After reviewing the needed properties that are being used, you will need to dig into your workflows to understand a number of items:

- Does business logic exist? If so, what is its purpose?
- Does the workflow talk to external systems? If so, how?
- Does the workflow use the vRA7 plugin?
- Is information being written back to vRA? If so what information?

While every situation is unique, and you may have workflows that can be adapted to work with vRA8 with some simple modification, you should strongly consider starting over and recreating these workflows specific to vRA8. Your existing workflows were designed and architected around the vRA7 architecture, which is significantly different from that of vRA8. While things make it look like a good idea in principle, they may not work out so well in practice.

How SovLabs can help you start preparing for vRealize Automation 8 today.

We've been working with many organizations to start preparing for vRA8 by displacing the aforementioned customizations with out of the box software tools. One by one, we're removing these items today so you don't have to re-write, modify, or recreate these items when you are ready to make the move to vRA8.

To learn more about how we can help with your migration to vRA8 and the benefits of moving away from customizations on vRA7 in preparation for vRA8, please attend our upcoming webinar or contact us for a personalized review of how we can help you prepare for what's coming in vRA8.

SovLabs vRealize Automation Optimization and Upgrade Assessment

While businesses expect improved flexibility and control with vRealize Automation 8 (vRA 8), many don't anticipate the significant level of effort required to upgrade.

Built on a completely new code base, moving to vRA 8 allows organizations to modernize their infrastructure automation and deliver at the speed of DevOps. However, like other IT infrastructure, existing vRA environments have become highly customized over time and many of these customizations do not easily 'lift and shift' to vRA 8.

SovLabs makes future proofing your vRA investment simple. Whether you're looking to move to vRA 8, or just want to find ways to reduce complexity, SovLabs vRealize Automation Optimization and Upgrade Assessment can provide insight into environmental areas that present an opportunity for optimization.

The assessment will look at key data points and interdependencies to help:

- Identify your current environmental conditions, including the number of blueprints, reservations, and software components - all representing potential areas of optimization.
- Admins understand the level of effort to maintain and add new services to existing environments.
- Highlight migration considerations for vRA 7.x to vRA 8.x upgrade.