SovLabs vRealize Automation Optimization Analysis Findings

Date: May 12, 2020

Report Content:

Executive Summary

SovLabs engaged with X on May 12, 2020 to perform an analysis of the current vRealize Automation (vRA) and associated platforms and applications utilized to deliver private cloud to X. This engagement consists of gathering environmental data across tenants and providing a consolidated view of all vRA environments along with potential optimizations that could benefit your environment.

The SovLabs Analyzer Tool examines more than 18 critical variables and their interdependencies which will determine the level of effort to:

[1] Maintain the current environment
[2] Add new services to the current environment and
[3] Upgrade the current vRA 7.x environment to version 8.1 or 8.2.

Upon analysis of the data, it appears that Software Components are critical to the success of the vRealize Automation platform for the X. With Software Components not currently available in the vRA 8 platform, this could introduce significant risk to the vRA 8 migration process. While some of the use cases could be met with cloud-init or cloudbase-init, it would likely take a significant amount of time to transfer all the required software components over to the new platform.

Adopting the SovLabs modules today can help X better prepare for the vRA 8 migration by taking a policy-based approach to machine provisioning while reducing the amount of custom code in the environment. This can mitigate risk associated with having to potentially retool almost 21,000 lines of custom code to make it work with vRA 8.

Analyzer Tool Findings

Environment Blueprint Findings

Name	Custom Naming
Total Blueprints	61
Non-Reducible Blueprints	57
Potential Final Blueprints	58

61 blueprints is considered a medium number of blueprints, it is recommended that this number be reduced. Having more than 40 blueprints increases the number of admins required to maintain your automation environment. This will also increase the difficulty and time to migrate from vRA 7 to vRA 8.

After reviewing the blueprints in your environment we have estimated that we can potentially help reduce your blueprints down to 58. This number excludes 57 blueprints that are not candidates for reduction. Blueprints that include canvas based items such as Software Components (42), Nested Blueprints (15), Multi-Machine blueprints (2), and certain NSX components (3) are not candidates, however we can further review these items to determine if there are other possible design options.

Developing a standardized Blueprint strategy is a key element to reducing complexity and configuration overhead within vRealize Automation 7 environments. Having a standardized blueprint consumption model can assist with reducing other elements in the environment such as Business Groups, Reservations, External Network Profiles, as well as reduce custom code for integrations. It also allows for standardized consumption from external sources such as ServiceNow, Terraform, API, and other solutions.

Resource Placement Findings

Resource Placement Elements	Number of Elements
Total Reservations	186
Total Business Groups	62
Reservation Policies	7
Storage Reservation Policies	6
Datacenter Locations	0

To further optimize vRealize Automation 7 and build on the reduction in needed Blueprints, SovLabs recommends reducing the number of Business Groups which effectively helps reduce the number of managed allocation-based Reservations. This is especially important for organizations looking to move to the new vRealize Automation 8 platform. vRealize Automation 8 no longer has the concept of reservations making it a truly consumption-based platform; this is significantly different than vRealize Automation 7. Natively within vRealize Automation 7 the reservation concepts are allocation-based which can cause scale and management challenges.

Many organizations today are looking to implement consumption-based models that are in line with the model available in vRealize Automation 8. This is very challenging to do natively in vRealize Automation 7 when many business groups are in use. We have found a total of 186 reservations and 62 business groups in your environment. That is an average of 3 reservations per business group.

By adopting a consumption-based model the total number of business groups can be reduced, ultimately reducing the number of managed reservations. This is in line with the methodologies available in vRealize Automation 8 which no longer utilizes the reservation concept. The SovLabs modules include vRealize Automation 7 enhancements such as the SovLabs Property Toolkit and Template Engine that can help reduce the need for maintaining many business groups and reservations within vRealize Automation 7. This aids in reducing management overhead as well as further simplifying the migration to vRealize Automation 8.

Custom Code Findings

Elements using Custom Code	Number of Elements
Total Lines of Custom or Community Written Code	20912
XaaS Blueprints	66
Event Broker Subscriptions	39

Approval Policies using Event Subscription	11
Dynamic Drop Downs	11
Custom Resource Actions	0
Potential Reduced Lines of Code	8365

Many organizations have taken a custom code first approach to integrating vRealize Automation in their environments to meet their unique requirements. This type of an approach causes the following operational challenges:

- Use of unsupported customizations in a production environment
- Difficult to keep up with Endpoint API changes
- Upgrades are markedly more complex
- The skill set required to develop the solution
- Costly increase in technical debt

Our recommendation is to minimize custom code as much as possible within your environment, more of a custom code last type of approach. This could include better utilization of the available constructs in vRealize Automation or taking a policy driven approach to provisioning using the fully supported suite of SovLabs products.

Application Optimization Findings

After reviewing the number of blueprints in the environment utilizing Software Components, we have determined that an additional 40 blueprints may possibly be eliminated. This could potentially bring your total blueprint count down as low as 18 helping to further reduce management overhead and achieve a higher level of standardization.

Software Components are canvas based blueprint components, causing the blueprints that use them to be very static and inflexible in nature. Blueprints using Software Components are therefore incapable of being consolidate d without taking a different approach. In order to consolidate these blueprints, the scripts that make up the Software Components need to be executed dynamically. This can be accomplished by utilizing tools like Ansible Tower, Puppet, or the SovLabs LifeCycle Components Toolkit.

Using one of these tools, the execution of scripts can be attached to requests dynamically allowing a single blueprint to effectively deploy many different applications. This will significantly reduce the management overhead associated with managing additional static blueprints. This approach also prepares your application provisioning scripts for migration to vRealize Automation 8 as application provisioning no longer uses Software Components, but now utilizes Cloud-Init or Cloudbase-Init.

Network Optimization Findings

Elements using Custom Code	Number of Elements
External Network Profiles	485
NAT Network Profiles	0
Routed Network Profiles	15
Private Network Profiles	0

Your environment currently includes 485 external network profiles. To effectively achieve a reduction in blueprints and reservations it is also important to reduce the number of external network profiles. To achieve a consumption based model as discussed earlier, all networks need to be available from any reservation. This would amount to having 90210 touch points to properly configure your environment. Manual configuration of these touch points often leads to errors and configuration drift within your environments, and even missing one of these touchpoints can lead to request failures that are difficult to troubleshoot.

Configuration touchpoints are not just created during the initial setup and configuration of vRealize Automation. As networks become fully utilized there is ongoing management a nd operational overhead in determining how to add new networks and remove full networks that can prove to be particularly challenging.

Using the Sovlabs IPAM module we can effectively drive the number of external network profiles down to 0 and remove all the configuration overhead associated with managing the network associations in all of your 186 reservations. The SovLabs IPAM module can be driven dynamically using business logic to ensure the proper network placement for workloads while also significant ly reducing configuration overhead. Our IPAM approach also allows for additional reduction in management overhead by eliminating much of the ongoing management as it regards to adding new networks and dealing with networks that have become fully allocated.

Appendix A Summary Data

Environmental Summary - Total for all environments

Element	Number of Elements
Approval Policies	14
Azure Reservations	2
Business Groups	62
Composite Blueprints	61
Endpoints	6
Entitlements	290
Event Broker Subscriptions	39
Machine Prefixes	600
Network Profiles	500
Property Definitions	53
Property Definitions Defined by vRO Action	11
Property Groups	29
Reservation Policies	7
Software Components	64
Storage Reservation Policies	6
XaaS Blueprints	66
vSphere Compute Resources	14
vSphere Reservations	184

Environmental Summaries - Totals

Tenant URL: Redacted 81g/RF9MmCTYEmMN1unpyacMXEhlUX3BA/zJC6KMKBA=/vcac/org/EDC by Tenant

Element	Number of Elements
Approval Policies	14
Azure Reservations	2
Business Groups	62
Composite Blueprints	61
Endpoints	6
Entitlements	290
Event Broker Subscriptions	39
Machine Prefixes	600
Network Profiles	500
Property Definitions	53
Property Definitions Defined by vRO Action	11
Property Groups	29
Reservation Policies	7
Software Components	64
Storage Reservation Policies	6
XaaS Blueprints	66
vSphere Compute Resources	14
vSphere Reservations	184

Appendix B - Blueprint Mappings

This section gives insights into the type of canvas elements used when authoring blueprints in your environment. The counts are representative of how many blueprints are using at least one of these elements. These findings are listed as a summary for all tenants.

Component Type	Count	Blueprint Names
XaaS Blueprint	12	Removed
vSphere (vCenter) Machine	45	Removed
Software Component	42	Removed
Composite Blueprint	3	Removed
On-Demand Security Group	1	Removed
On-Demand Routed Network	3	Removed
Existing Security Group	2	Removed
On-Demand Load Balancer	1	Removed
App.Container.Request	1	Removed
Container Network	1	Removed

Appendix C - Definitions

Component	Definition
Approval Policies	Approval policies define the governance for a request and its provisioned resources. Approval policies could be simplistic in that all requests go to a particular group to be approved, or they could be much more complex in nature, even allowing the use of custom coded vRO workflows to generate approval for the request.
Business Groups	A business group associates a set of services and resources to a set of users, often corresponding to a line of business, department, or other organizational unit.
Composite Blueprints	Composite blueprints, or rather, blueprints are the construct that allow an organization to define what a deployment, and its request form looks like. Blueprints increase in complexity as more elements are added to the canvas, as software components are included, when nesting other blueprints or XaaS blueprints, or in the sheer number of blueprints (blueprint sprawl) present in an environment. With an increasing number of blueprints, manageability of the solution becomes increasingly difficult; particularly when a lot of the blueprints are handling similar functions.
Event Broker Subscriptions	EBS, or Event Broker Subscriptions are used to call custom coded vRealize Orchestrator workflows at predefined event stages in vRA. Due to the requirement to write custom code to take advantage of EBS, these can add quite a bit of complexity to an environment. When multiple EBS are in use for a single blueprint, the complexity compounds because of potential dependencies between the EBS workflows. A ny time custom code is introduced into a solution, it increases complexity in an environment, increase technical debt for your organization, can be difficult to support from an operations perspective, and it can increase risk when migrating or upgrading to a new version of the product.
Entitlements	Entitlements grant permissions to specified users of a business group to catalog services, items, or 2nd day actions on provisioned resources.

Component	Definition
Naming Prefixes	Naming prefixes allow you to define a hostname standard in vRealize Automation. vRA 7 default naming prefixes are very static in nature in that a prefix cannot be dynamically determined based off of user submitted inputs. If a naming prefix is added to a blueprint, that naming prefix is the only name valid for the resources. In addition, vRA does zero validation within the environment to validate that the name is, in fact, unique, and can lead to potential name duplication and/or failed requests.
Property Groups	A Property Group allows for the inclusion of specific custom properties into a group to be able to more easily apply multiple custom properties simultaneously.
Reservations	Reservations are the elements in vRA 7 that allow members of a Business Group access to provision resources on specific Compute Resources (vSphere Clusters, AWS Region, etc.), storage, and networks, and constrain the amount of each resource available to the business group. Reservations increase in complexity with an increase in the number of network profiles in use in each reservation, the number of business groups in your environment, and the number of both reservation policies and storage reservation policies in use in your environment.
Reservation Policies	Reservation policies are used to control how reservation requests are processed. When you provision machines from the blueprint, provisioning is restricted to the resources specified in the selected reservation policy. Reservation Policies, in and of themselves, are not overly complex, they are fairly simple to set up and apply to your reservations. However, a large number of Reservation Policies in your environment indicates that there is resource placement logic that points to an increase in complexity in the environment as a whole. In addition, the only two ways to select a Reservation Policy in vRA is either by exposing them to a user (can lead to users having to make a selection they do not understand), or by writing complex custom code.

Component	Definition
Resource Actions	Day 2 resource actions are custom coded actions designed to take action against an existing vRA catalog resource (Deployment, Virtual Machine, etc.). Any time custom code is introduced in to a solution, it increases complexity in an environment, increase technical debt for your organization, can be difficult to support from an operations perspective, and it can increase risk when migrating or upgrading to a new version of the product.
Software Components	The construct that vRealize Automation 7 uses for application provisioning. vRA Enterprise licensing is required to use this feature, and each of your templates require the installation of an agent to use this feature. In addition, software components are typically very static in nature, and can lead to blueprint sprawl.
Storage Reservation Policies	Storage Reservation policies are used to control how reservation requests are processed from the perspective of storage. When you provision machines from the blueprint, provisioning is restricted to the storage resources specified in the selected storage reservation policy. Storage Reservation Policies, in and of themselves, are not overly complex, they are fairly simple to set up and apply to your storage resources. However, a large number of Storage Reservation Policies in your environment points to complex resource placement logic that points to an increase in complexity in the environment as a whole. In addition, the only two ways to select a Storage Reservation Policy in vRA is either by exposing them to a user (can lead to users having to make a selection they do not understand), or by writing complex custom code.
XaaS Blueprints	XaaS (Anything as a Service) Blueprints are custom coded vRealize Orchestrator workflows that can be presented from the vRealize Automation catalog. While some XaaS Blueprints can be fairly simple in nature, more often these types of blueprints are used to integrate very complex business logic in to the provisioning process, or to create custom resources that are not already defined in vRA. The level of customization and complexity in any environment goes up substantially when numerous XaaS Blueprints are in use. Any time custom code is

Component	Definition
XaaS Blueprints (cont.)	introduced in to a solution, it increases complexity in an environment, increase technical debt for your organization, can be difficult to support from an operations perspective, and it can increase risk when migrating or upgrading to a new version of the product.