

Hybrid Cloud: Deploy to Azure Using VMware vRealize Automation

An Analysis of Benefits and Limitations

By Tony Wu and Simon Alexander
Global AzureCAT

April 2020

Contents

Introduction	3
About this e-book	3
About vRA and vRO	4
vRA methods for Azure deployment	4
XaaS blueprints for Azure method	5
Use the XaaS blueprints for Azure method to create a blueprint	6
Limitations in this approach	15
Azure Resource Manager template via XaaS method	16
Advantages of this method	16
Disadvantages of this method	17
Use the Resource Manager template via XaaS method to create a blueprint	17
Conclusions	21
Learn more	21
Appendix A: Use the default VMware Azure Machine blueprint	22
Limitations in this approach	22
Performance issues	22
Scale limits	23
Cost and billing issues	24
Appendix B: Test scenarios and results	25
Appendix C: Install vRealize Automation on Azure	31

List of figures

Figure 1. End-to-end capabilities across the vRealize Suite	4
Figure 2. vRA and vRO workflow	5
Figure 3. Completed blueprint in the vRealize Automation catalog	15
Figure 4. The newly created "arm demo" blueprint in the vRealize Automation catalog	20
Figure 5. Server deallocation and billing costs with the default template	24
Figure 6. Using Azure VMware by CloudSimple to run vRA on Azure	31

Authored by Tony Wu and Simon Alexander. Edited by Nanette Ray. Reviewed by Global AzureCAT.

© 2020 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Introduction

Many organizations have made significant investments in VMware technologies for their datacenters and processes, including deployment, orchestration, cost management, and governance. However, VMware has recently introduced new capabilities in their cloud deployment tooling that may limit organizations that use Microsoft Azure services.

These capabilities include the new VMware Cloud Automation Service, a software-as-a-service (SaaS) option for cloud infrastructure and application delivery. In addition, the rewrite of VMware vRealize Automation (vRA) in version 8.0 extends cloud capabilities to on-premises environments. However, the initial release of the vRA rewrite does not support Azure Resource Manager templates, lacks certain governance support, and does not currently provide a migration path from version 7.6 to version 8.0.

At Microsoft, we work with many organizations that take a deliberate, hybrid approach to the cloud. They want to integrate the cloud model within the context of their existing VMware ecosystem. A hybrid cloud approach provides important benefits when your organization is in transition in its journey to the cloud. It allows you to begin to incorporate cloud services within your existing tooling and processes—steps that help mitigate risks, reduce costs, and lessen the pressure on change management systems while also providing ongoing flexibility.

Many organizations with an investment in VMware also use the VMware vRealize Suite as the primary management plane. They use it to manage deployment, orchestration, costs, and governance of their virtual machine (VM) and cloud ecosystem. VMware recognizes the value of hybrid cloud capabilities and has introduced features that support deployment to cloud providers, including Azure—notably, through its vRealize Automation (vRA) and vRealize Orchestration (vRO) applications.

If your organization wants to integrate Azure into your existing vRealize capabilities, this e-book is for you. We focus primarily on vRealize Automation (vRA) to show you how to get the most out of its deployment capabilities and support a largely complete integration with Azure. We offer recommendations for working with vRA version 7.6 that will help you move forward in your cloud migration strategies while waiting for the improvements in version 8.0 or until a migration path becomes available.

About this e-book

This e-book explores the capabilities of the three main methods to deploying on Azure and discusses the advantages and disadvantages of each method. Our goal is to make sure you have access to the Azure services and configuration options you need to protect your current and future technology investments.

The recommendations in this e-book can help you maximize the breadth of Azure services that you can support from vRA as well as minimize the time to availability of new and future Azure services—steps that traditionally required vRA upgrades to achieve.

About vRA and vRO

vRealize Automation (vRA) consists of a self-service dashboard, capabilities to build out blueprints for infrastructure resources (or combinations of resources), and the ability to create integrations with major on-premises and cloud-based industry infrastructure endpoints. Essentially, vRA allows you to define your infrastructure, and it surfaces functionality that allows you to provide more of a self-service experience to your users.

vRealize Orchestrator (vRO) is the workflow engine that allows you to extend the capabilities within vRA to allow for more elaborate and custom interactions. vRO can be called from other applications, such as vRealize Operations Manager, and has full REST capabilities to call out to external resources. The focus of vRO is workflow, and it works like a multi-tool when you need to address more complex fulfillment scenarios.

The two products work together with other VMware products to provide the capabilities shown in Figure 1.

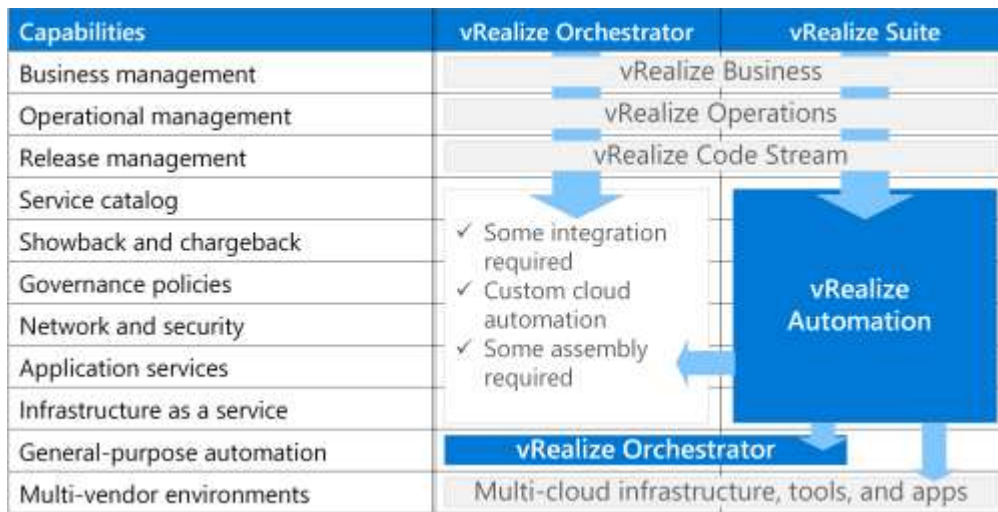


Figure 1. End-to-end capabilities across the vRealize Suite

vRA methods for Azure deployment

In vRA, there are three distinct methods you can use to deploy to Azure:

- The VMware anything-as-a-service (XaaS) blueprint method.
- The Azure Resource Manager template method as supported by the VMware XaaS blueprint capability.
- The VMware Azure Machine blueprint using the default Azure VM template. This method provides basic support for VM deployments to Azure, but we do not recommend it because of the limitations that are discussed in [Appendix A](#).

The first two methods—XaaS and Resource Manager template via XaaS—offer libraries of workflows that you can use, unlike the default Azure Machine template. This e-book focuses on the first two methods. For most workloads, these two methods give you the best results.

XaaS blueprints for Azure method

Using the XaaS blueprints for Azure method, you can access and build workflows for most Azure services and capabilities, which are supported through the underlying Azure API. While greater training and knowledge is required to successfully deploy using this method as compared to the VMware Azure VM blueprint method, the learning curve is more than made up for by the breadth of capability.

This method relies on vRO, which provides the backend workflow deployment engine integrated in vRA. The vRO server that is distributed with vRealize Automation is preconfigured. When a system administrator deploys the vRealize Automation Appliance, the vRO server is also set up and starts running as the following figure shows.

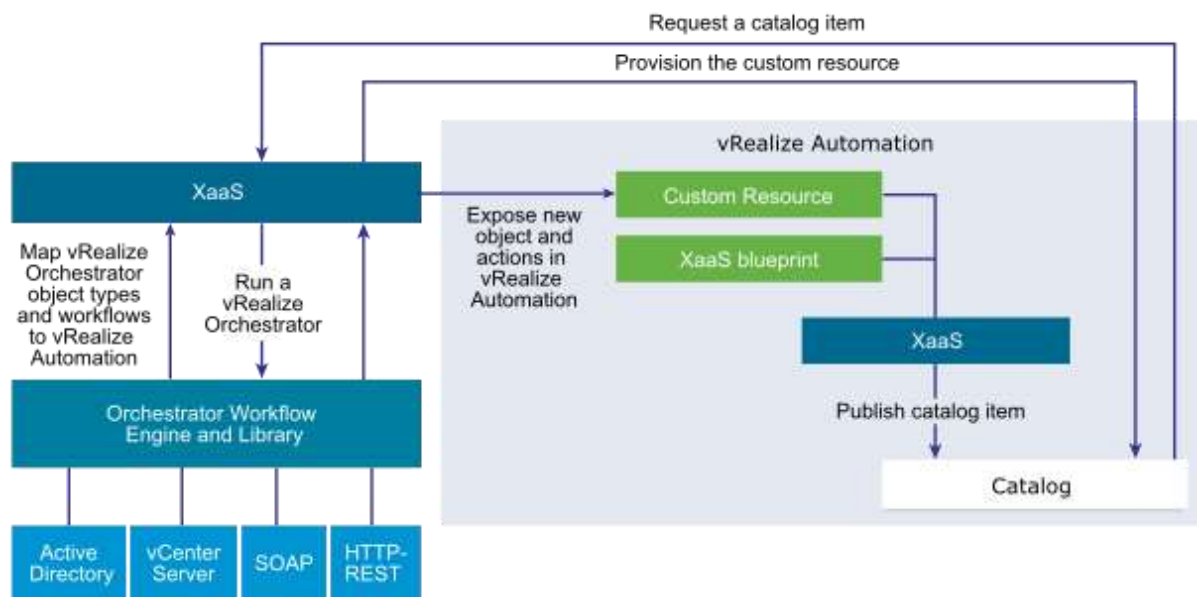


Figure 2. vRA and vRO workflow

To add custom resources, you use a built-in library of workflows to create an XaaS blueprint. This blueprint becomes the complete specification of a deployed workload solution that users can easily deploy. XaaS uses vRealize Orchestrator to execute the blueprint-defined workflow by communicating with the Azure control plane service to provision cloud resources.

Compared to the default Azure Machine template workflow, the XaaS library workflow supports more software lifecycle operational actions for the disks, network, resource groups, and VMs.

However, the XaaS library workflow presents a few disadvantages. For example, you can't deploy to new or existing virtual machine scale sets or choose from the full range of VM sizes. For more information, see [Limitations in this approach](#) later in this guide.

NOTE: For details about limitations by feature, see [Appendix B: Test scenarios and results](#).

Use the XaaS blueprints for Azure method to create a blueprint

This section describes how to create a simple, IaaS-based VM deployment on Azure that is defined by an XaaS-based Azure machine blueprint template. We call this the XaaS blueprints for Azure method.

A successful deployment is a three-step process:

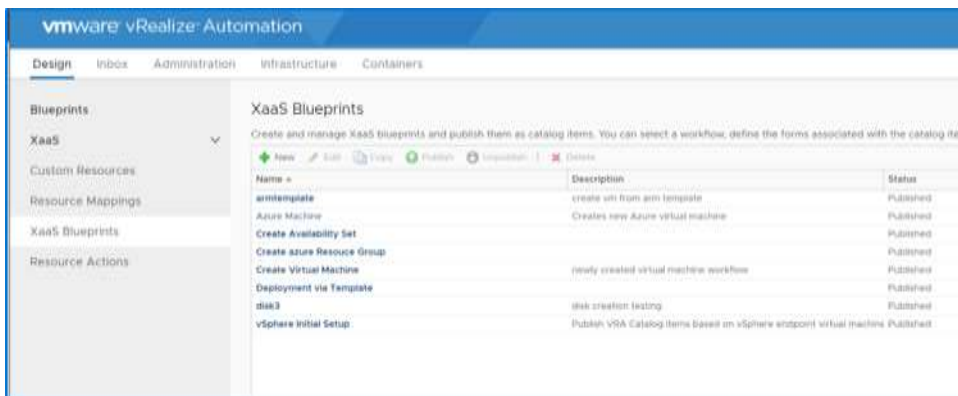
1. Create a blueprint template from a XaaS-based Azure machine blueprint template.
2. Create and publish your blueprint using the blueprint template.
3. Enable a user to deploy a workload using the blueprint.

Each step is described in the following sections.

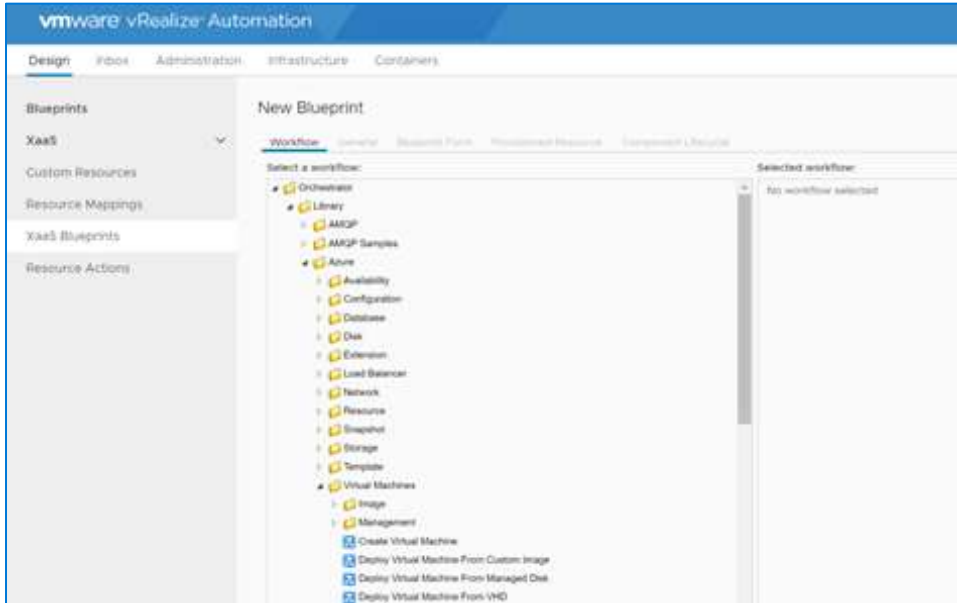
Create a blueprint template

The first step is to create a blueprint for an Azure workload. You start from the **XaaS Blueprints Design** tab of vRA using an XaaS-based machine template from the XaaS library of workflows. The steps are as follows:

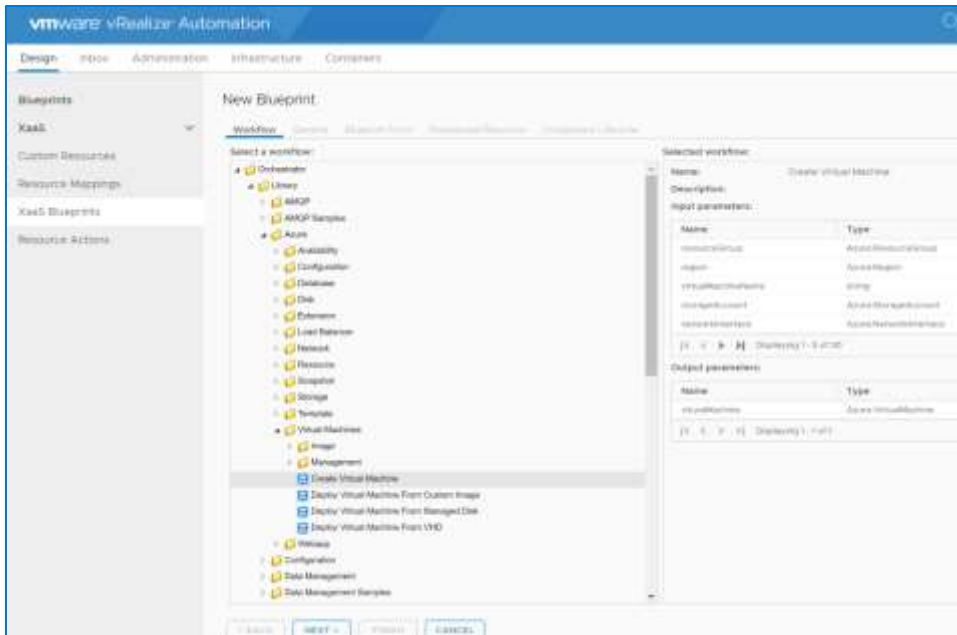
1. On the **Design** menu, click **XaaS Blueprints** and select the + **New** button.



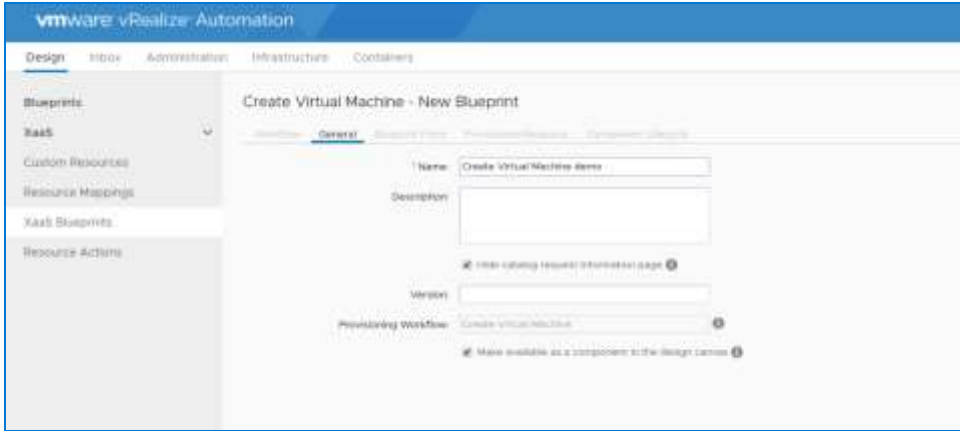
2. Within the **Library** folder, go to **Azure > Virtual Machines > Management** to display the Azure VM options.



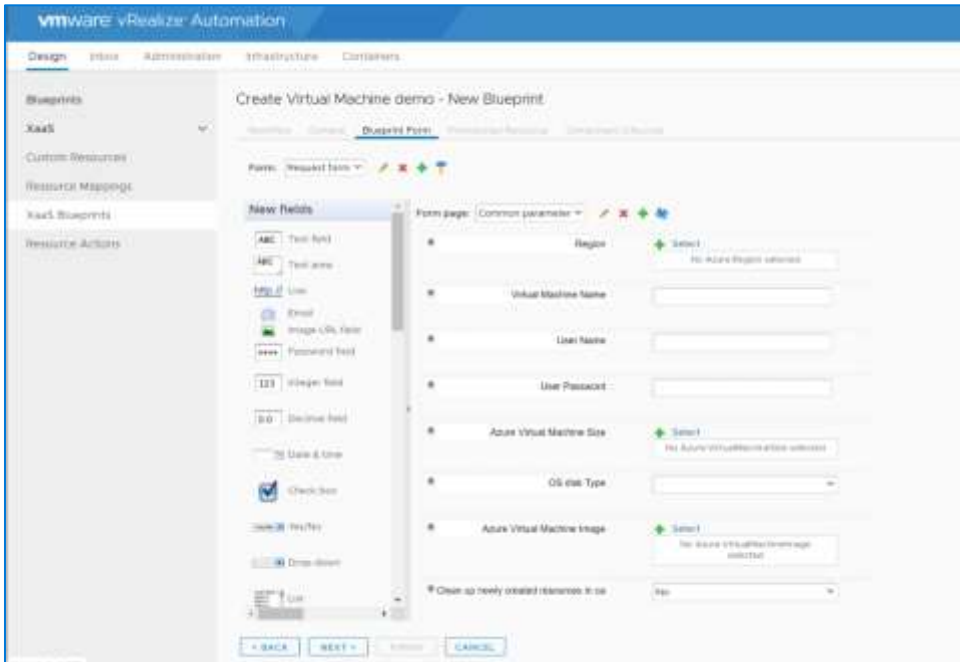
3. Select **Create Virtual Machine**, and then click the **Next** button to define the VM requirements.



4. In the **General** tab of the **Create Virtual Machine – New Blueprint** window, fill in the details according to your requirements. This step creates a template in the design canvas, so it is important to name the blueprint according to your organization’s naming practices. Ensure that the **Make available as a component in the design canvas** box is checked, and then click **Next**.



5. In the **Blueprint Form** tab, define the policies that will determine the options that users can select when deploying their VMs. For example, you can specify the regions that the users can choose, the Azure VM image, and other options. Complete this form with your organization's deployment policies in mind, then click **Next**.



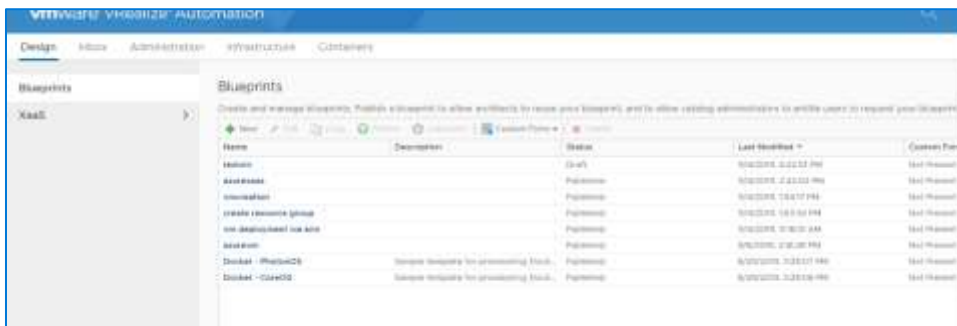
9. Select the **Create Virtual Machine Demo** item and click **Publish**. Notice that **Status** changes to **Published**, indicating that the blueprint is now available for use in the next sequence.



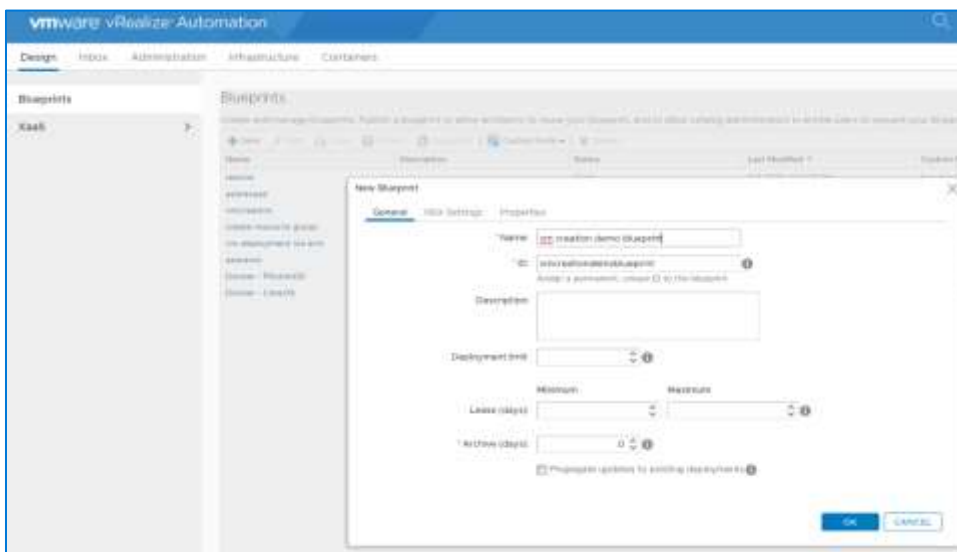
Create and publish a blueprint

The next step is to create a vRA composite blueprint that utilizes your new machine blueprint template to then make the blueprint available for users to deploy.

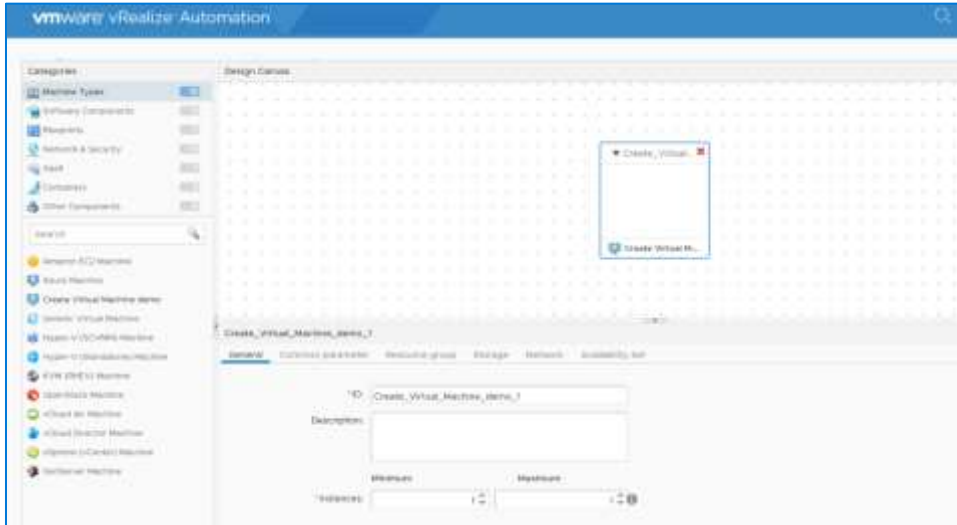
1. Under **Design**, click **Blueprints** and select the **+ New** button.



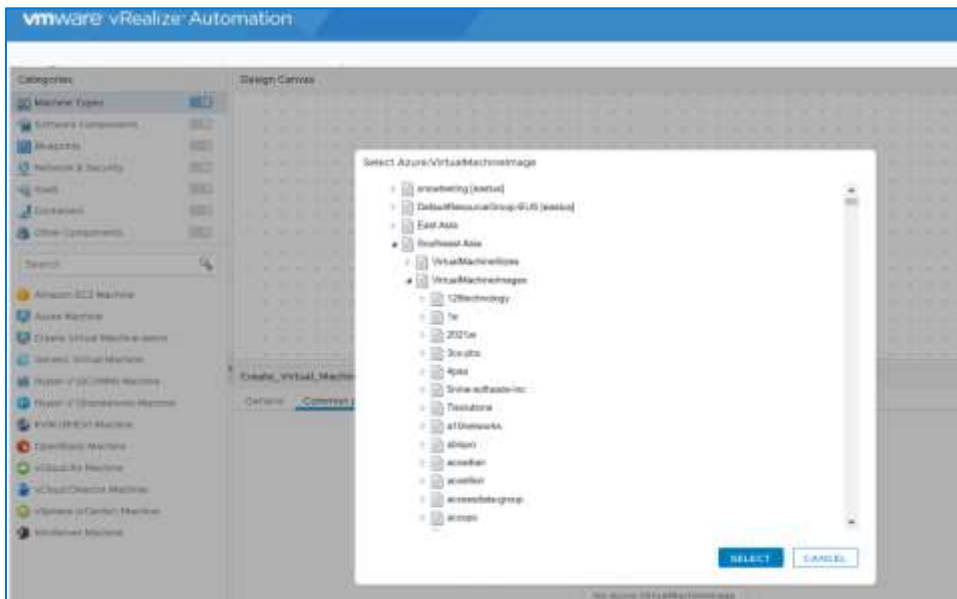
2. Provide a blueprint name and ID, then click **OK**.



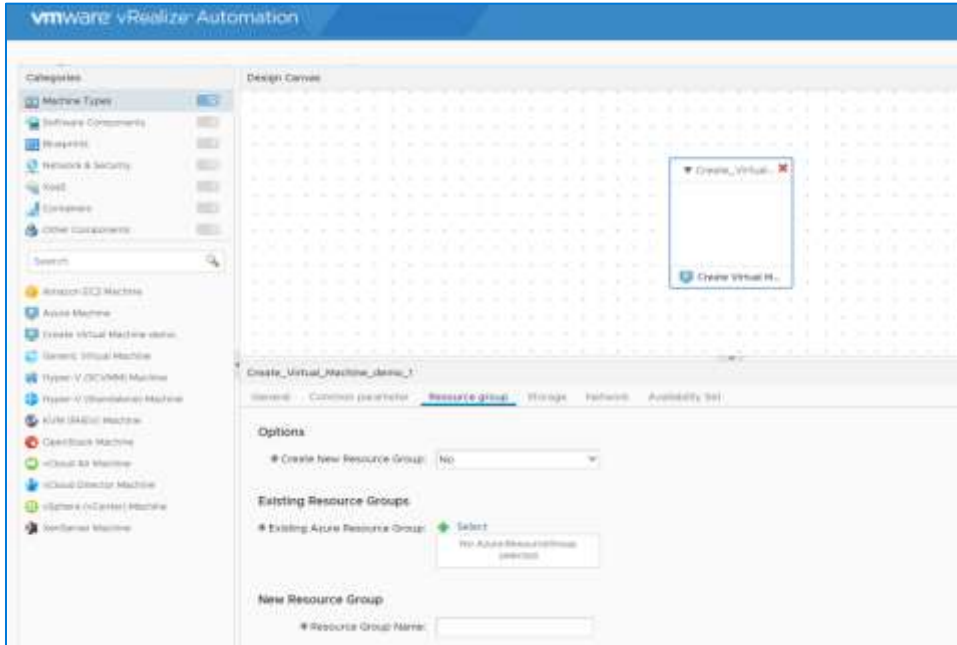
- In the **Design Canvas** screen that appears, notice the **Create Virtual Machine demo** that you created earlier in the lower-left window. Drag this blueprint template into the center window to start the design of the composite blueprint. Enter the required parameters for the workflow items starting with the **General** tab, where you specify the workflow constraints and options available to the users when they deploy.



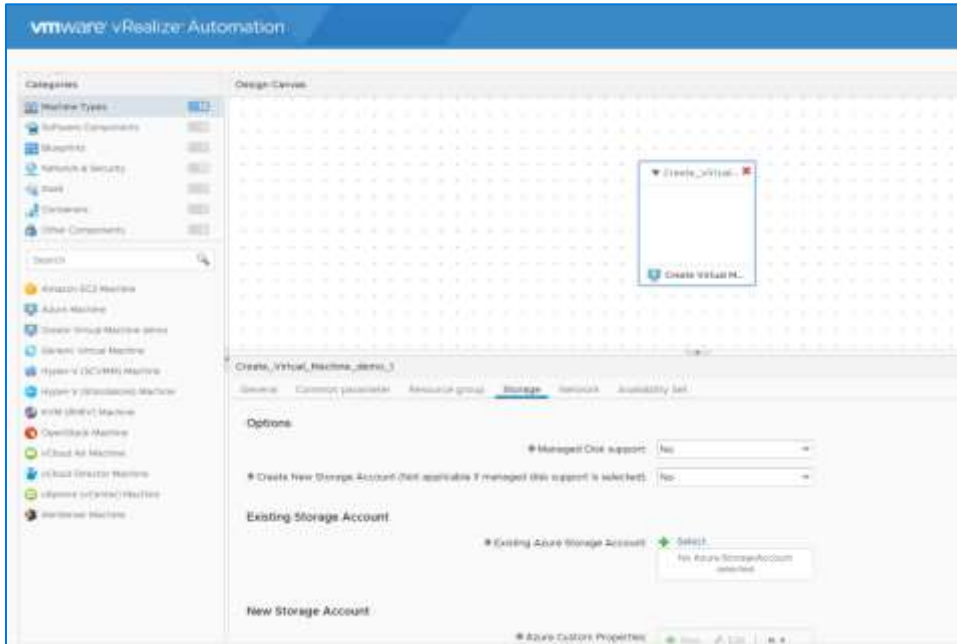
- Choose the **Select** button next to **Azure Virtual Machine Image** to display the following menu, where you can choose a machine image from the library.



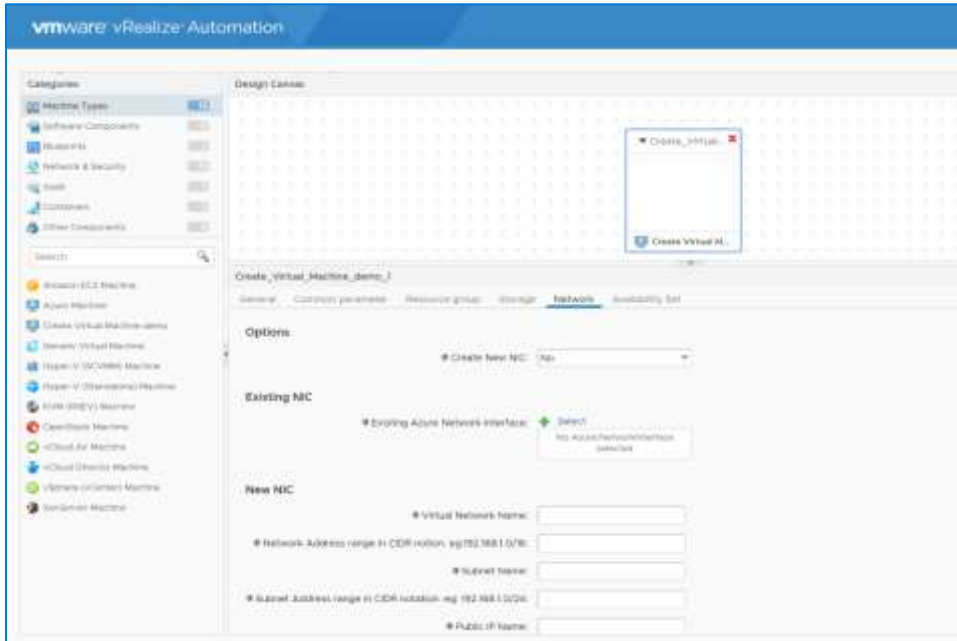
5. Click the **Resource Group** tab and configure either a new or existing resource group to deploy into.



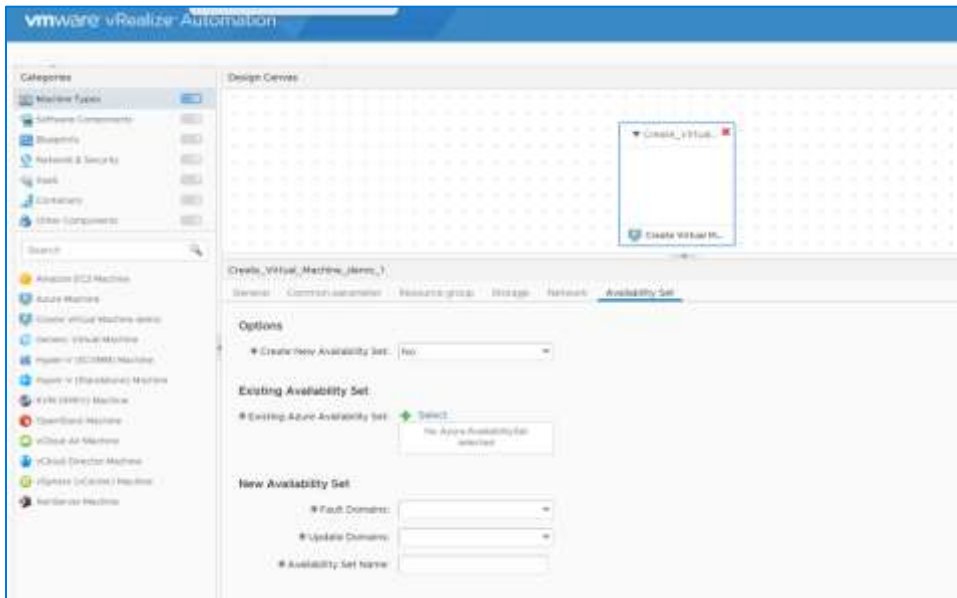
6. Click the **Storage** tab to configure options such as **Managed Disk support**, a new or existing storage account to deploy to, and the storage type.



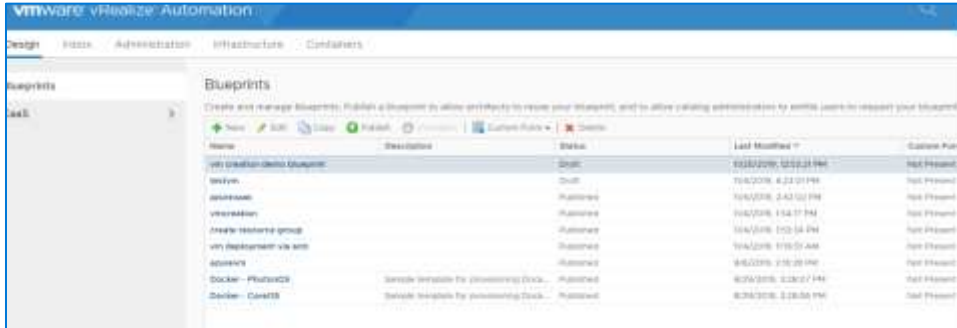
7. Click the **Network** tab and configure all the network options.



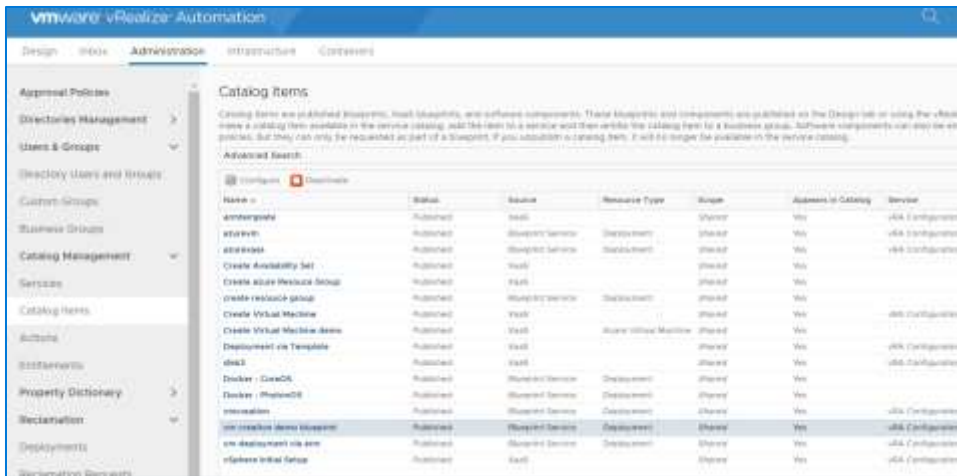
8. Click the **Availability Set** tab and configure the availability sets.



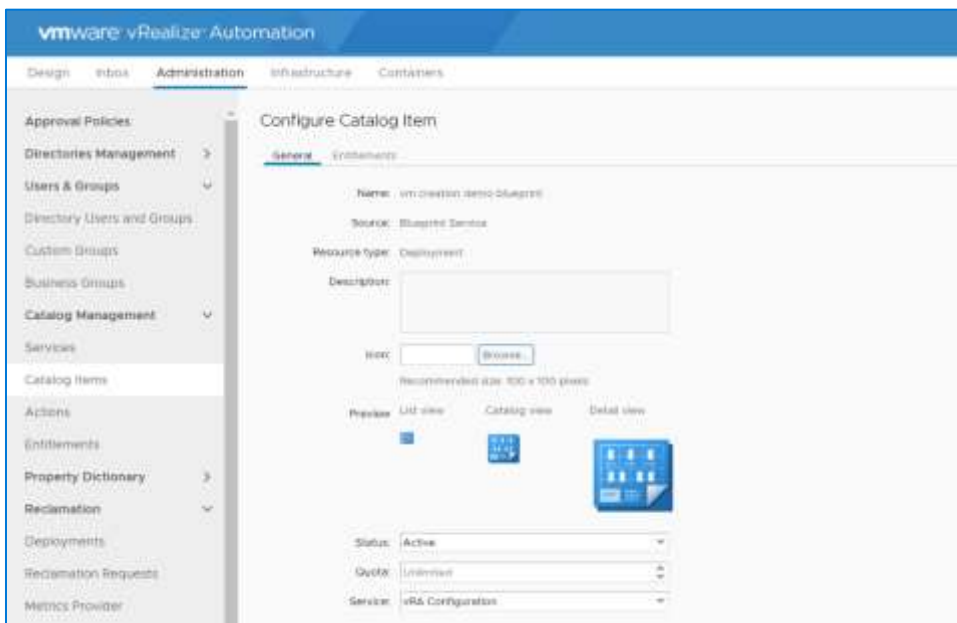
- When all the configuration items are complete, select **Finish** to complete the blueprint draft. Then, to publish the blueprint, click the **Publish** button.



- Click **Administration** in the menu bar and select the published blueprint.



- Set the status to **Active** to enable it in the vRealize Automation configuration service and make it available to users for deployment.



Enable end-user deployment

The final step is to ensure that users can now see your newly created blueprint in the catalog, which means it's ready for deployment.

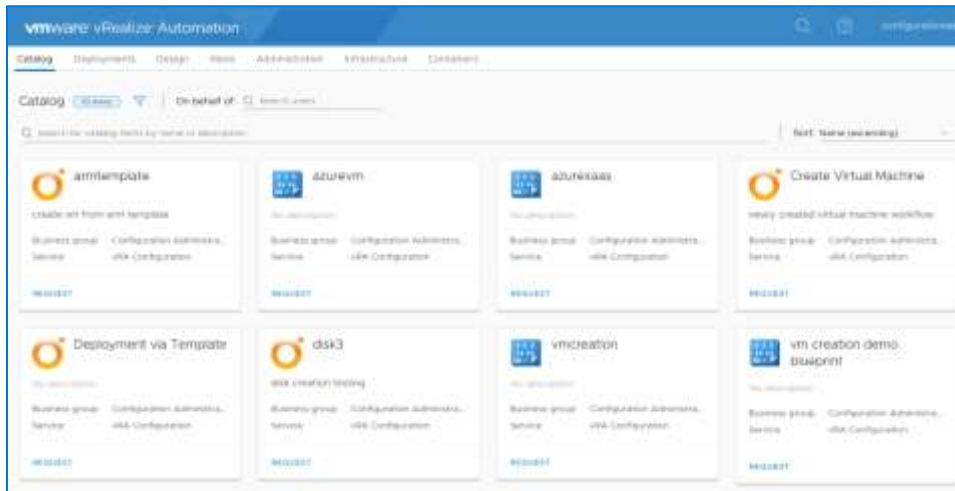


Figure 3. Completed blueprint in the vRealize Automation catalog

Limitations in this approach

Using the VMware XaaS blueprints for Azure method, you can create a deployment template that includes the most common requirements, such as Premium disks, Azure Managed Disks, and availability sets. This method supports the full set of VM sizes and families.

However, be aware of the following limitations:

- Some new and recently released features—for example, Ultra SSD—are not supported automatically without an upgrade to vRA and vRO.
- Whitelisted Azure regions are not selectable.
- It's not clear how deployments will work with existing governance tools.
- In lifecycle operational support, the actions that administrators can take on deployed resources may be limited.

You can avoid most of these limitations related to Azure features by using Azure Resource Manager to create XaaS blueprints for Azure as described in the next section.

Azure Resource Manager template via XaaS method

Instead of using the XaaS blueprints for Azure method to create a deployment template, you can use the Azure Resource Manager template via XaaS method. Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in an Azure subscription. Management features such as access control, locks, and tags can be used to secure and organize resources after deployment.

For an overview of this service, see [Azure Resource Manager template overview](#) in the Azure documentation.

To implement infrastructure as code for your Azure solutions, you use Resource Manager templates. The template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax, which lets you state what you intend to deploy without having to write the sequence of programming commands to create it. In the template, you specify the resources to deploy and the properties for those resources. For details, see [Azure Resource Manager template deployment](#).

VMware vRO provides a template workflow that you can use to deploy Azure cloud services via Resource Manager templates. This XaaS workflow makes use of native Resource Manager templates to communicate with Azure Resource Manager.

This approach has several benefits. For example, you can define any service in Azure and deploy it as part of a workflow or blueprint. In addition, Resource Manager templates are a standard deployment tool for Azure. You can reuse these templates for any toolset that deploys to Azure.

Resource Manager provides many other usability benefits, such as:

- ✓ Declarative syntax.
- ✓ Repeatable results.
- ✓ Orchestration.
- ✓ Built-in validation.
- ✓ Modular files.
- ✓ The ability to create any Azure resource.
- ✓ Tracked deployments.
- ✓ Policy as code.
- ✓ Deployment blueprints.
- ✓ Continuous integration (CI) and continuous deployment (CD) integration.
- ✓ Exportable code.
- ✓ Authoring tools.

Advantages of this method

Resource Manager templates are integrated with XaaS blueprints in vRA, so you can provision the full menu of Azure native services using the supported features available to Resource Manager. You don't encounter the gaps observed with other workflows included in the Azure XaaS plug-in.

Resource Manager templates are a standard deployment mechanism for Azure. Any templates you create can be used in other deployment tools that can make an API call to Resource Manager. In addition, you can take advantage of the wide variety of Resource Manager templates that are

available in the GitHub marketplace—many of which need little to no customization.

Resource Manager templates also give you a level of abstraction from your toolset. This means that there is no lost IP or rework if you want to move to another deployment tool. As new services are released, you can use Resource Manager templates to deploy them without needing to upgrade your deployment tool.

Resource Manager templates are also tool-agnostic. You can select from a library of existing templates in the GitHub marketplace, and you can port templates to other management platforms.

Disadvantages of this method

As with any new service, there is a learning curve to adopting Resource Manager templates. In addition, some enterprises simply prefer not to use a central UI to execute their cloud lifecycle management tasks, such as operations, monitoring, troubleshooting, and recovery configurations.

Use the Resource Manager template via XaaS method to create a blueprint

You can create an XaaS blueprint by starting from the built-in Resource Manager template workflow in the vRO Azure library.

With this method, you can copy the entire contents of the main template and associated parameter template from any of the Azure-based solutions. You have full access to the native features and support provided by Resource Manager templates. You can also provide the Resource Manager main and parameter template URIs without copying the content.

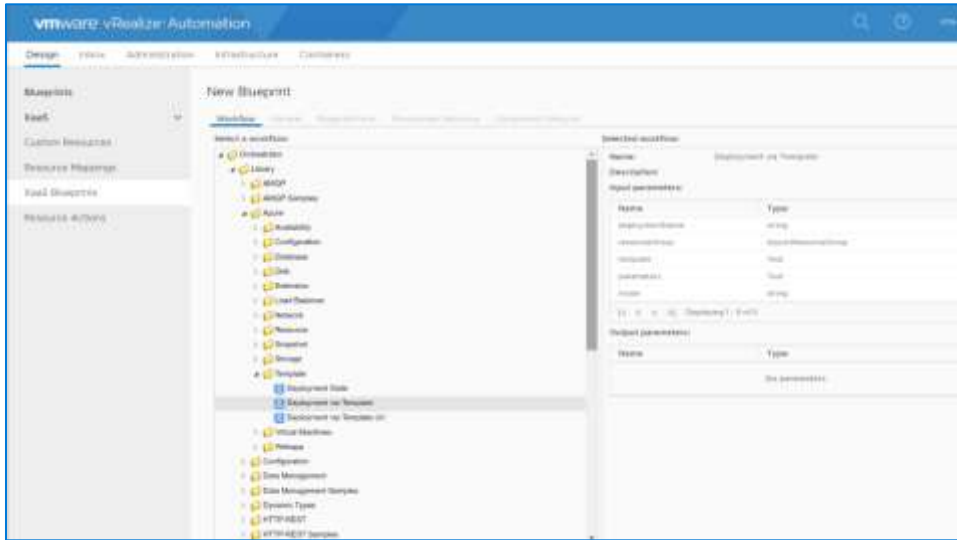
The following example shows you how to use this method to create a simple, IaaS-based Azure VM deployment that is defined by an XaaS-based Azure machine blueprint template.

NOTE: This guide assumes a working knowledge of Resource Manager templates and syntax. The steps focus on how to deploy a pre-existing Resource Manager template via vRA.

1. Under XaaS Blueprints, click **+ New**.



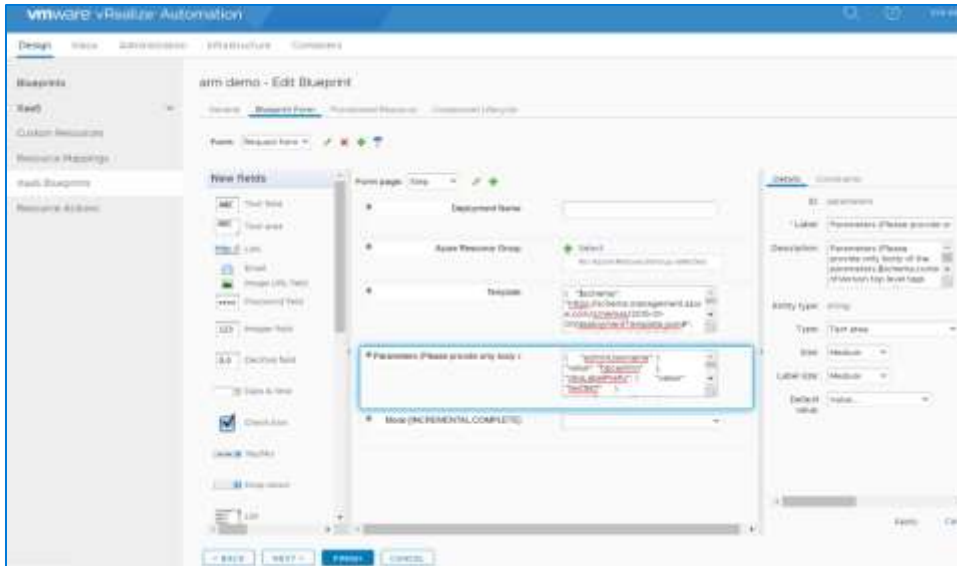
- Expand **Azure > Template** to select the **Deployment via Template** option.



- Click **Next** and define the name of the blueprint.



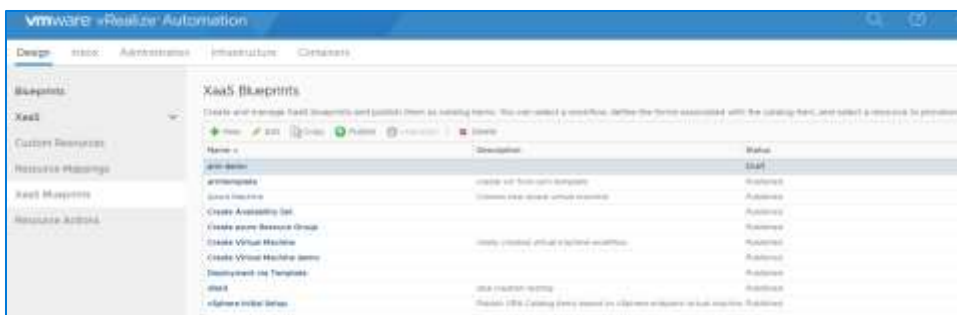
- Go to the **Blueprint Form** tab and do the following:
 - In the **Template** area, enter the Resource Manager template JSON main contents.
 - Under **Parameters**, enter the template parameters JSON contents. Note that only the body of the parameter is needed as shown.



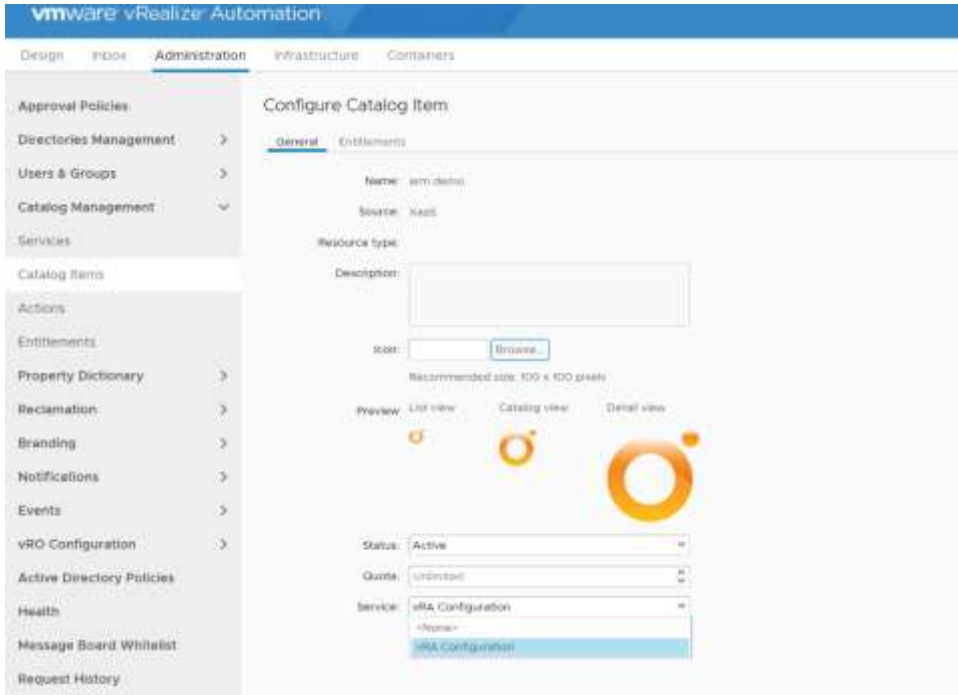
5. Click **Next** and review the configuration in each tab until you have reached the final step. Then select **Finish**.



6. To publish the template, click the **Publish** button.



7. Click **Administration** in the menu bar and go to **Catalog Management > Catalog items**. Click the name of the blueprint you just created. Make sure the **Status** is set to **Active**. Under **Service**, select **vRA Configuration** from the drop-down list.



The newly created blueprint is now available and appears in the catalog, ready for users to deploy.

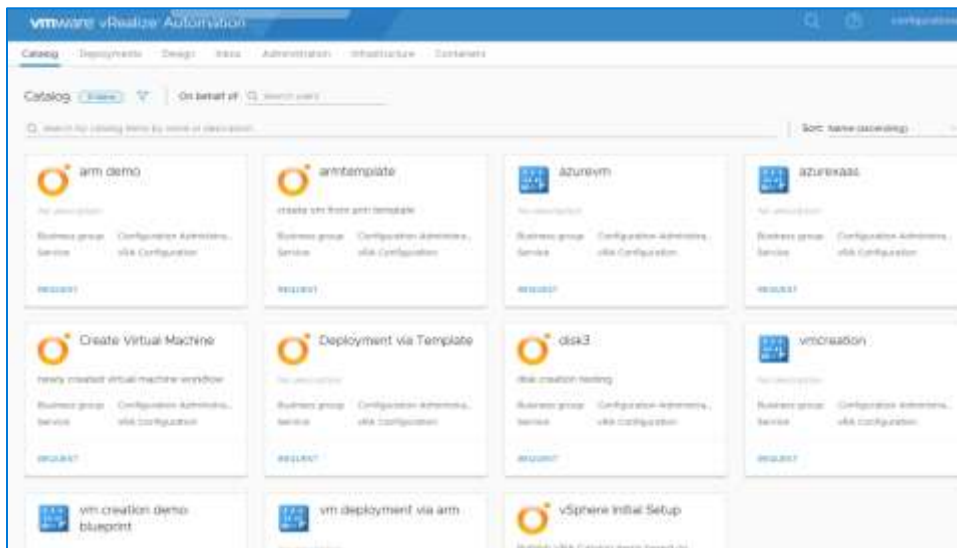


Figure 4. The newly created "arm demo" blueprint in the vRealize Automation catalog

Conclusions

The two XaaS methods of deploying Azure resources represent significant improvements in capability and flexibility compared to deployment via the default Azure Machine blueprint. In general, we do not recommend using the Azure Machine blueprints option for most workloads, because significant features are not available.

Here are our recommendations:

- To meet the requirements of most basic workloads, use the library workflows provided by XaaS for the deployment of Azure platform services.
- To include Azure features that are not supported by the standard XaaS method—Ultra SSD, for example—use the XaaS blueprints for Azure method.
- For complex workloads made up of multiple components and services, use the Azure Resource Manager template via XaaS method.
- Use Azure portal for enterprise-wide, lifecycle, and day-to-day operational management of the Azure services that support the solution workload.

Learn more

For more information about working with Azure, see the following resources:

- [Azure Quickstart Templates on GitHub](#)
- [Azure Resource Manager template overview](#)
- [Azure Resource Manager template deployment](#)
- [VMware documentation](#)
- [VMware Azure default virtual machine blueprint](#)
- [Azure VMware solution](#)



Go to the Azure portal at <https://portal.azure.com/>

Appendix A: Use the default VMware Azure Machine blueprint

You can use the default Azure Machine template option in vRealize Automation as the basis of an Azure blueprint. It contains the basic parameters you need to create a VM in Azure, such as compute, storage, and network. However, it has limited support of Azure compared to the XaaS Azure library workflows. For details, refer to [vRealize Automation documentation](#).

To enable the full use of all Azure cloud services in vRA, it is best to take advantage of the XaaS Azure library workflow integrated with Resource Manager templates to address some of the gaps mentioned earlier.

Limitations in this approach

The default Azure Machine template is limited in ways that impact the viability of most production deployments:

- Storage capacity and performance scalability are limited to up to 4 disks each with a maximum size of 1 TB.
- Compute scalability is limited to the set of VMs that were available before the vRA version was released—that is, you don't have access to the very latest VM types.
- Network scalability is limited, because accelerated networking is not currently supported, and other networking features are limited to those that were available before the vRA version was released.
- The selection of other Azure cloud services is similarly limited to those available when the vRA version was released.

Performance issues

The default Azure Machine template option can introduce performance problems because of the following disk limitations:

- You can provision a maximum of only 4 disks.
- You can provision disks to a maximum size of only 1,023 GB.

Each disk can manage only a certain amount of IO and throughput. To build a platform with high IO and throughput demands, you must attach multiple disks to the host. Then the workload must be distributed across the disks—typically, by striping them. The IO and throughput are added with each disk until you hit the VM maximum throughput and IO limit. Therefore, the maximum IO and throughput is a combination of the disk type and VM size.

The following shows the throughput limits for Azure Premium SSDs. Smaller sizes (P1 to P10) are not shown. For a complete list, see [Azure VM disk sizes](#).

Table 1. Throughput limits for Azure Premium SSDs

Premium SSD sizes	P30	P40	P50	P60	P70	P80
Disk size in GiB	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	5,000	7,500	7,500	16,000	18,000	20,000
Throughput per disk	200 MiB/sec	250 MiB/sec	250 MiB/sec	500 MiB/sec	750 MiB/sec	900 MiB/sec

A limitation of 1,023 GB per disk allows you to deploy up to P30-sized disks. P40 sized disks and above are unavailable. This means that the maximum possible disk performance is 5,000 IOPS and 200 MiB/sec throughput.

In addition, because a maximum of 4 disks is the limit, performance is limited to 2,000 IOPS and 1,000 MiB/sec throughput.

Scale limits

The default Azure Machine template option can introduce scalability problems because of the following limitations in disk allocation:

- You can provision a maximum of only 4 disks.
- You can provision disks to maximum size of only 1,023 GB.

Platforms or applications requiring more than 4 x 1,023 GB (approx. 4 TB) storage cannot be hosted.

In Azure, high-scale VMs are available in different sizes with options for the number of CPU cores, memory, OS, and temporary disk size. Each VM size also has maximum number of data disks that you can attach to the VM. Therefore, the chosen VM size affects how much processing, memory, and storage capacity is available for your application. It also affects the compute and storage cost.

For example, the following table lists the specifications of the largest VM size in a DS series, DSv2 series, and a GS series.

Table 2. Example comparison of large VM sizes

VM size	CPU cores	Memory	VM disk sizes	Max. data disks	Cache size	IOPS	Bandwidth cache IO limits
Standard_DS14	16	112 GB	OS = 1,023 GB Local SSD = 224 GB	32	576 GB	50,000 IOPS 512 MB/sec	4,000 IOPS and 33 MB/sec
Standard_GS5	32	448 GB	OS = 1023 GB Local SSD = 896 GB	64	4224 GB	80,000 IOPS 2,000 MB/sec	5,000 IOPS and 50 MB/sec

The “Max. data disks” column shows you the limit on these VM sizes—32 and 64 disks, respectively. As the table shows, scale is significantly limited at 4 disks.

Cost and billing issues

To meet a specific solution requirement, the existing Azure machine templates require some complexities in manual customization, quality assurance, and final release. You are likely to incur extra cost and time when you use this method.

The default templates also introduce some operational limits in the way servers are shut down and deallocated, which typically adds to the cost. One of the major benefits to the cloud is the ability to turn off services when not in use without incurring billing costs. See the following article for more details: [Virtual machines lifecycle and states](#).

To see how this works, the following figure shows a server’s states. A server continues to be billed until it is considered deallocated. vRA can power down a server, but it can’t move it into the deallocated state. So, the server continues to be billed.

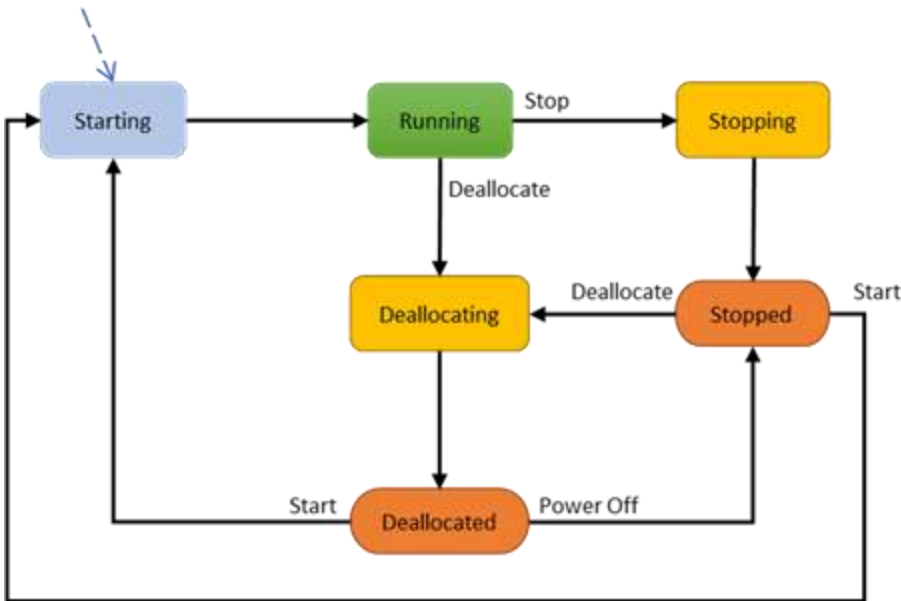


Figure 5. Server deallocation and billing costs with the default template

Appendix B: Test scenarios and results

To compile the information and recommendations in this document, we designed and executed several tests to explore the capabilities and limitations within each deployment method within vRA. See the following table for the list of tests we conducted.

Table 3. Tests conducted

Test	Test scenarios
Ability to deploy VM SKUs	Are all SKUs available? If not, what SKUs are available?
Management test cases	Tagging, resource groups, and stop/deallocate/start scenarios.
Image deployment: Public vs. private images	Can you deploy marketplace images or private SOE images?
Region, availability, and scalability scenarios	Are all regions available? Can you utilise scale sets and availability zones? Is Azure Site Recovery available?
Disk and storage options	Are all the disk types available? Are there limitations?
Load balancing	What is available for use?
Monitoring and recovery	What monitoring and management options are available?

This document highlights many of the key findings, but for a complete list of test results and supported services, see the following table.

Table 4. Test results and supported services

Test case	Azure VM blueprints	XaaS blueprints for Azure method	ARM template method
VM sizes			
A-series	✓	✓	✓
Av2-series	✓	✓	✓
Bs-series	✓	✓	✓
D-series	✓	✓	✓
Dv2-series	✓	✓	✓
Dv3-series	✓	✓	✓

DS-series	✓	✓	✓
DSv2-series	✓	✓	✓
DSv3-series	✓	✓	✓
Ev3-series	✓	✓	✓
ESv3-series	✓	✓	✓
F-series	✓	✓	✓
FS-series	✓	✓	✓
FSv2-series	✓	✓	✓
H-series	✓	✓	✓
LS series	✓	✓	✓
LSv2 series	✗	✓	✓
M-series	✓	✓	✓
Mv2-series	✗	✓	✓
NC-series	✓	✓	✓
NCv2-series	✓	✓	✓
NCv3-series	✓	✓	✓
ND-series	✓	✓	✓
NV-series	✓	✓	✓
NV-series	✓	✓	✓
NVv2-series	✓	✓	✓
NVv3-series	✓	✓	✓
G-series	✓	✓	✓
GS-series	✓	✓	✓
Whitelisted sizes	✗	✗	✓

Management test cases			
Add existing tag	✗	✓	✓
Add new tag	✗	✓	✓
Deploy into existing resource group	✓	✓	✓
Deploy into new resource group	✗	✓	✓
Images			
Deploy using public images	✓	✓	✓
Deploy using private images	✓	✓	✓
Scenarios—region, availability, and scalability			
Able to deploy to all public regions	✗	✓	✓
Able to deploy to whitelisted regions	✗	✗	✓
Deploy to existing scale set	✗	✗	✓
Deploy to new scale set	✗	✗	✓
Deploy to existing availability set	✓	✓	✓
Deploy to new availability set	✗	✓	✓
Deploy to Availability Zones	✗	✗	✓
Set up Azure Site Recovery	✗	✗	✓
Disk types			
Create Standard HDD	✓	✓	✓
Create Standard SSD	✗	✗	✓
Create Premium SSD	✓	✓	✓
Create Ultra SSD	✗	✗	✓
Create Managed Disks	✓	✓	✓

Create Ephemeral Disks	✗	✗	✓
Maximum disks (disk count)			
2 (e.g. B1s)	✓	✓	✓
4 (e.g. D4s_v3)	✓	✓	✓
8 (e.g. D8s_v3)	✗	✓	✓
16 (e.g. D16s_v3)	✗	✓	✓
32 (e.g. D32s_v3)	✗	✓	✓
64 (e.g. DS5_v2)	✗	✓	✓
Maximum disk sizes supported			
32 GiB (e.g. P4/E4/S4)	✓	✓	✓
64 GiB (e.g. P6/E6/S6)	✓	✓	✓
128 GiB (e.g. P10/E10/S10)	✓	✓	✓
256 GiB (e.g. P15/E15/S15)	✓	✓	✓
512 GiB (e.g. P20/E20/S20)	✓	✓	✓
1 TiB (e.g. P30/E30/S30)	✓	✓	✓
2 TiB (e.g. P40/E40/S40)	✗	✓	✓
4 TiB (e.g. P50/E50/S50)	✗	✓	✓
8 TiB (e.g. P60/E60/S60)	✗	✓	✓
16 TiB (e.g. P70/E70/S70)	✗	✓	✓
32 TiB (e.g. P80/E80/S80)	✗	✓	✓
64 TiB (e.g. Ultra Disk)	✗	✗	✓
Networking scenarios			
Deploy to existing virtual network (VNet)	✓	✓	✓
Deploy to new VNet	✗	✓	✓

Deploy to existing subnet	✓	✓	✓
Deploy to new subnet	✗	✓	✓
Deploy with public IP address	✗	✓	✓
Deploy with no public IP address	✓	✓	✓
Deploy with inbound ports enabled (80,443,22,3389)	✗	✓	✓
Deploy with no inbound ports enabled	✓	✓	✓
Deploy with existing NSG	✓	✓	✓
Deploy with new NSG	✗	✓	✓
Deploy with Accelerated Networking enabled	✗	✗	✓
Create new route and route table	✗	✓	✓
Load balancing			
Deploy without load balancing	✓	✓	✓
Deploy with Azure Application Gateway Standard	✗	✗	✓
Deploy with Application Gateway Standard v2	✗	✗	✓
Deploy with Application Gateway WAF	✗	✗	✓
Deploy with Application Gateway WAF v2	✗	✗	✓
Deploy with Azure Basic Load Balancer	✗	✓ *	✓
Deploy with Azure Standard Load Balancer	✗	✓	✓
* Only internal load balancer and external load balancer			
Monitoring and recovery			
Enable detailed monitoring into workspace	✗	✗	✓
Enable boot diagnostics with existing storage account	✓	✗	✓

Enable boot diagnostics with new storage account	✗	✗	✓
Enable OS guest diagnostics	✗	✗	✓
Enable Azure Managed Identity	✗	✗	✓
Enable auto-shutdown	✗	✗	✓
Enable auto-shutdown notifications	✗	✗	✓
Enable backup to existing Azure Site Recovery vault	✗	✗	✓
Enable backup to new Azure Site Recovery vault	✗	✗	✓
Deploy with existing backup policy	✗	✗	✓
Deploy with new backup policy	✗	✗	✓
Other			
Ability to deploy with extensions	✗	✓	✓
Ability to deploy with cloud-init script	✗	✗	✓
Ability to deploy to a dedicated host	✗	✗	✓

Appendix C: Install vRealize Automation on Azure

You can run vRealize Automation VMware from an existing on-premises VMware deployment. However, you may choose to deploy this infrastructure to Azure, whether for testing or production purposes. However, VMware vRealize Automation cannot run directly on a native Azure VM as this setup is not officially supported by Microsoft and VMware.

Instead, you can deploy to a native VMware ESX cluster in Azure to support the vRA automation. You must use [Azure VMware by CloudSimple](#) to do this.

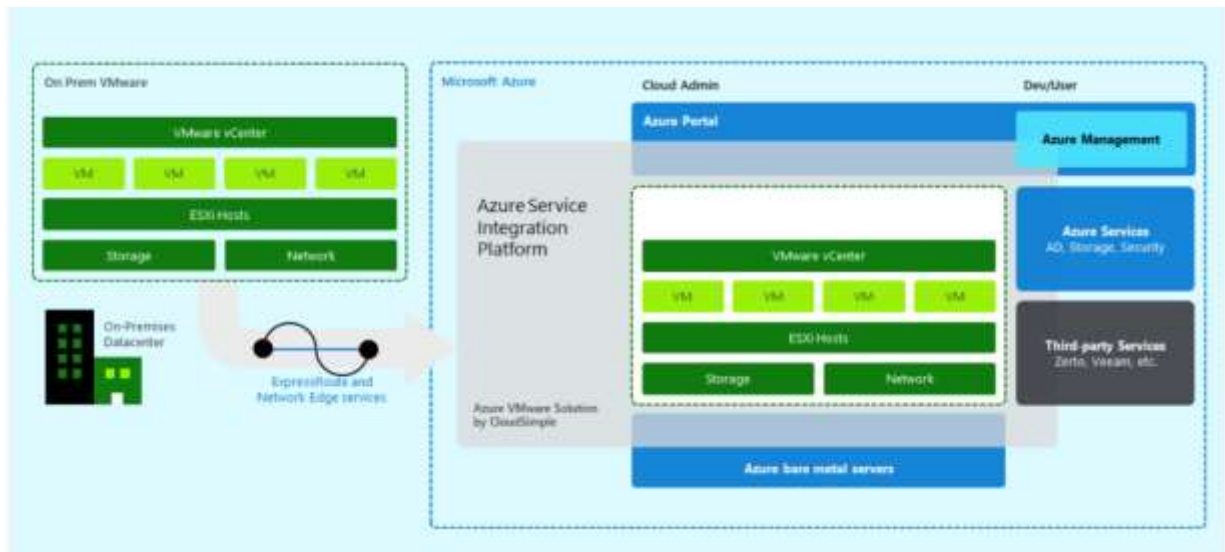


Figure 6. Using Azure VMware by CloudSimple to run vRA on Azure

There are several benefits to hosting the vRealize automation tools on an Azure VMware environment. Compared to hosting on-premises, this hybrid cloud setup—VMware private and Azure public cloud—does the following:

- The communication path between the vRealize tools and public Azure cloud API provider is shorter with low latency network (collocated), thus further speeding up the deployment of Azure cloud services.
- Extending the VMware environment to Azure makes its integrated management tools and native services available for flexible application deployment. You get a dedicated, isolated, high-performance infrastructure plus unique Azure products and services.