



Professional Expertise Distilled

# Mastering vRealize Automation 6.2

Enable virtualization, automation, and efficient cloud management  
in your environment with vRealize Automation 6.2

**J. Powell**

**[PACKT]** enterprise   
PUBLISHING professional expertise distilled

[www.allitebooks.com](http://www.allitebooks.com)

# Mastering vRealize Automation 6.2

Enable virtualization, automation, and efficient cloud management in your environment with vRealize Automation 6.2

**J. Powell**

**[PACKT]** enterprise   
PUBLISHING professional expertise distilled

BIRMINGHAM - MUMBAI

# Mastering vRealize Automation 6.2

Copyright © 2015 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: September 2015

Production reference: 1250915

Published by Packt Publishing Ltd.  
Livery Place  
35 Livery Street  
Birmingham B3 2PB, UK.

ISBN 978-1-78217-377-9

[www.packtpub.com](http://www.packtpub.com)

# Credits

**Author**

J. Powell

**Project Coordinator**

Shipra Chawhan

**Reviewers**

Omar Abdulrahman Ghazal

Karthic Kumar

Yannick Struyf

**Proofreaders**

Stephen Copestake

Safis Editing

**Commissioning Editor**

Kunal Parikh

**Indexer**

Hemangini Bari

**Acquisition Editor**

Nikhil Karkal

**Graphics**

Disha Haria

**Content Development Editor**

Mamata Walkar

**Production Coordinator**

Komal Ramchandani

**Technical Editors**

Dhiraj Chandanshive

Gaurav Suri

**Cover Work**

Komal Ramchandani

**Copy Editor**

Sonia Michelle Cheema

# About the Author

**J. Powell** is an active proponent of VMware's virtualization technologies for small businesses and nonprofits, with a focus on Horizon View and vCenter. He is a senior system engineer at Nine Virtual Technologies (for more information, visit <http://www.ninevirtualtech.com>).

He has over a decade of IT experience in enterprise infrastructure design, implementation, and administration. He holds certifications from VMware and is active on their community forums.

He has previously worked for Packt Publishing in the capacity of technical reviewer for the book, *VMware Horizon View Essentials*, which was published in October 2014.

In addition to his IT experience, he has owned and operated a fine dining restaurant in North Carolina, Fifty Fifty Lounge and Grill. He currently resides in Indianapolis, IN, with his beautiful wife, Dena, and their daughter, Parker.

# About the Reviewers

**Omar Abdulrahman Ghazal** is a dedicated and solutions-focused IT professional who offers more than 10 years of experience spanning the operation, delivery and installation, consulting, design, presales, project management, sales, and business projection of cutting-edge technical solutions that exceed clients' needs in the IT infrastructure solutions and technologies including, but not limited to, cloud computing, virtualization and datacenter management, identity and security, and unified communication. He is an analytical technology and business negotiator as well as a problem solver with a proven track record of completing complex projects within time constraints. A detail-oriented team player, he has an extensive cross-functional experience working within cross-boundary communication and collaboration. He possesses solid interpersonal, communication, management and decision-making skills. He has the proven capability of delivering business value propositions around IT infrastructure solutions to ITDMs and BDMs. He has achieved and exceeded specified commitments, including securing pilots, assessments, and revenue. He has demonstrated expertise in strategies, plans, and tactics to sell technology solutions.

---

First of all, I would like to thank "ALLAH". I'd also like to thank my beloved family, my daughter, Jana, and everyone who has helped me on this book.

---

**Karthic Kumar** works as a senior cloud solutions architect at vCloud Air and has been with VMware since 2010. He handles the design and implementation of complex customer scenarios within the vCloud Air environments. He also serves as an escalation point for the resolution of demanding technical issues that customers experience when using vCloud Air. He is also involved in the designing of the training curriculum for the other employees and customers.

He spends his spare time on researching various cloud products and services, cloud orchestration, and automation products. He also writes his own blog, <http://www.virtualizationexpress.com>. He holds several certifications, including VCP, MCP, VCAP-DCA, and VCAP-CIA. He was also recognized as a VMware vExpert for two consecutive years (2014 and 2015).

His Twitter handle is @carthick1. He can also be found on LinkedIn at <http://in.linkedin.com/in/carthick1>.

**Yannick Struyf** is a cloud integration associate working for Nubera, a VMware Enterprise solution provider in Benelux. When he started his career, he quickly identified his passion for infrastructure and virtualization. With a background and strong focus on systems and networking, he specializes in automation, orchestration, and network virtualization, primarily with the VMware vRealize Suite and VMware NSX. He is currently working on several vRealize automation projects for Nubera and VMware Professional Services in Belgium and Luxembourg. Yannick is a VMware certified professional, and he holds a master's degree in applied informatics from KU Leuven.

# www.PacktPub.com

## Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit [www.PacktPub.com](http://www.PacktPub.com).

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

## Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

## Free access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

## Instant updates on new Packt books

Get notified! Find out when new books are published by following @PacktEnterprise on Twitter or the *Packt Enterprise* Facebook page.





# Table of Contents

<b>Preface</b>	<b>vii</b>
<b>Chapter 1: Introducing vRealize Automation 6.2</b>	<b>1</b>
<b>Introduction to vRealize Automation</b>	<b>1</b>
Benefits of vRA	2
Beginning of vRealize Automation 6.2	3
<b>Available editions of vRealize Automation 6.2</b>	<b>4</b>
vRealize Automation Standard edition	6
vRealize Automation Advanced edition	6
vRealize Automation Enterprise edition	7
<b>Summary</b>	<b>7</b>
<b>Chapter 2: vRealize Automation Architecture and Feature Overview</b>	<b>9</b>
<b>What edition should I choose?</b>	<b>10</b>
<b>vRealize Automation core components</b>	<b>10</b>
vRealize Automation Identity Appliance	11
vRealize Automation Appliance	11
IaaS server	11
<b>High-level architecture overview</b>	<b>13</b>
<b>Features of vRealize Automation 6.2</b>	<b>14</b>
Integration with vROps	15
CloudClient	16
Pay-as-you-go support	17
<b>Other features improved in this release</b>	<b>17</b>
Blueprints	18
Build profiles	18
Clone deployment	18
WIM deployment	18
SCCM deployment	19

Kickstart deployment	19
Infrastructure Organizer	19
Associating costs to resources	20
Logs for auditing and compliance	20
vRealize Orchestration (vRO)	21
Integration with VMware Horizon View desktops	21
Tenants and SSO authentication	21
<b>Summary</b>	<b>23</b>
<b>Chapter 3: Designing and Building a vRealize Automation 6.2 Infrastructure</b>	<b>25</b>
<b>Proving the technology</b>	<b>26</b>
<b>Proof of Concept</b>	<b>26</b>
<b>Proof of Technology</b>	<b>27</b>
<b>Pilot implementation</b>	<b>27</b>
<b>Designing the vRealize Automation architecture</b>	<b>29</b>
Design considerations	33
Naming conventions	33
Compute resources	33
Business and Fabric groups	33
Endpoints and credentials	34
Reservations	34
Reservation policies	35
Network profiles	35
Compute resources	35
Blueprints	35
Users and groups	36
Catalog management	37
Services	37
Catalog items	37
Entitlements	37
Actions	37
Approval policies	38
Advanced services	38
<b>Example design exercise</b>	<b>38</b>
vRealize Automation components	40
Calculating the amount of storage and compute resources	41
Network considerations	42
Naming the provisioned machines	42
Blueprints	43
Services and catalog items	43
<b>Summary</b>	<b>43</b>

<b>Chapter 4: Installing and Configuring vRealize Automation 6.2</b>	<b>45</b>
<b>Preparing for the installation</b>	<b>46</b>
What you need to get started	46
Downloading the software	47
<b>Installing the vRealize Automation Identity Appliance</b>	<b>48</b>
<b>Installing the vRealize Automation Virtual Appliance</b>	<b>49</b>
<b>Configuring the vRealize Automation Identity Appliance</b>	<b>50</b>
<b>Configuring the vRealize Automation Appliance</b>	<b>52</b>
<b>Installing and configuring the IaaS server</b>	<b>55</b>
<b>Upgrading the vRealize Automation and Identity Appliances</b>	<b>63</b>
<b>Configuring vRealize Automation</b>	<b>64</b>
Configuring the vRealize Automation infrastructure	66
<b>Summary</b>	<b>77</b>
<b>Chapter 5: Mastering Blueprints</b>	<b>79</b>
<b>Creating a linked clone Blueprint</b>	<b>80</b>
Preparing the linked clone Blueprint for provisioning	84
Provision the linked clone Blueprint	86
<b>Creating a clone Blueprint</b>	<b>88</b>
<b>vRealize Automation guest agents</b>	<b>90</b>
<b>Creating a WIM Blueprint</b>	<b>91</b>
<b>Creating a Blueprint with MDT</b>	<b>93</b>
<b>Creating a SCCM Blueprint</b>	<b>96</b>
<b>Creating a Kickstart Blueprint</b>	<b>99</b>
<b>Multi-machine Blueprints</b>	<b>103</b>
<b>Provisioning physical machines</b>	<b>104</b>
<b>Property dictionary</b>	<b>104</b>
<b>Destroy a provisioned machine</b>	<b>106</b>
<b>Summary</b>	<b>107</b>
<b>Chapter 6: Creating Approval Policies</b>	<b>109</b>
<b>Getting started with Approval Policies</b>	<b>109</b>
Creating a Post Approval Policy	114
<b>Summary</b>	<b>114</b>
<b>Chapter 7: Installing and Configuring vRealize Automation Application Services 6.2</b>	<b>115</b>
<b>Downloading the vRealize Automation Application Services 6.2 software</b>	<b>117</b>
<b>Installing the vRealize Automation Application Services Appliance</b>	<b>118</b>
<b>Configuring the vRealize Automation Application Services Appliance</b>	<b>119</b>

Configuring the AppD web portal	121
Installing the AppD Bootstrap agent on a VM	123
Configuring SQL for provisioning	126
Summary	128
<b>Chapter 8: REST API and vRealize Orchestrator</b>	<b>129</b>
Introducing REST	130
Application programming interface (API)	132
vRealize Orchestrator configuration	133
Configure the vRO endpoint in vRA web portal	135
Running the vRealize Orchestrator Client	136
Navigating through vRealize Orchestrator Client	138
Defining our example customized workflow	140
Configuring the prerequisites in vRO	140
Creating a custom workflow with vRealize orchestrator client	149
Testing a customized workflow	164
Create a CPU and memory hot add workflow	165
Adding the workflows to vRA web portal	173
Additional workflow tools	174
Summary	174
<b>Chapter 9: Integrating vRealize Operations Manager</b>	<b>175</b>
Overview of vRealize Operations Manager 6.0	176
Benefits of integrating vROps with vRealize Automation	177
How to integrate vROps with vRealize Automation	177
Utilizing vROps to reclaim idle virtual machines	180
Summary	185
<b>Chapter 10: Customizing the End User Portal Experience</b>	<b>187</b>
Portal overview	187
How to make service requests	190
How to monitor service requests	192
Installing plugins to access VMs through the web portal	193
Modify previously deployed service requests	194
Customizing the end user web portal	195
Summary	196
<b>Chapter 11: Troubleshooting vRealize Automation 6.2</b>	<b>197</b>
Errors that may occur during installation and operation	197
Monitoring and logging	204
Getting further assistance	205
Summary	205

---

<b>Chapter 12: References for vRealize Automation 6.2</b>	<b>207</b>
Links within your vRA environment	207
VMware documentation	208
Other sources	231
<b>Index</b>	<b>233</b>

---



# Preface

Working with virtualization technologies like the ones under the VMware umbrella provide IT Administrators with the ability to manage more with less. What used to take many hours and hands now can be done in a fraction of the time and with less hands. As technology advances, we will enter an exciting time of convergence. This means we will have multiple services, applications, systems, and networks coming together and managed through a single pane of glass. vRealize Automation is the leading product in this field.

Before its Fall 2014 rebranding, vRealize Automation was known as vCloud Automation Center (vCAC). Its origins are based in a financial services company named Credit Suisse. The software was developed in 2005 to help the company roll out its virtualization strategy, while maintaining a focus on governance and operational control. After a few years of using the software in-house, to maintain thousands of virtual machines, Credit Suisse formed a separate company named DynamicOps to continue the development and innovation behind the software.

DynamicOps was launched at the beginning of 2008 in Burlington, Massachusetts. Leslie Muller, who led the charge in developing what would be known as vCAC, was the founder and CTO of DynamicOps. The company was acquired in 2012 by VMware.

The latest version of vRealize Automation is 6.2 and it will be the focus of this book. The product's strengths are also its weakness: Versatility. vRealize Automation has a lot of moving parts and components. It also takes a lot of time and consideration to configure all of the pieces. Throughout the book, you may question the case-sensitivity and spacing in the examples. They are not type-o's. Also, you will see references to DynamicOps as well as vCAC throughout the examples. Once you understand the basics, as well as some of the more advanced integration points, you will be well on your way to Mastering vRealize Automation.



## What this book covers

*Chapter 1, Introducing vRealize Automation 6.2*, defines vRealize Automation with a history of the product leading up to its current version, which is 6.2. It will also mention where the product is heading.

*Chapter 2, vRealize Automation Architecture and Feature Overview*, discusses the features and different servers that comprise vRealize Automation.

*Chapter 3, Designing and Building a vRealize Automation 6.2 Infrastructure*, discusses everything from POC to Production.

*Chapter 4, Installing and Configuring vRealize Automation 6.2*, including tips for a successful deployment.

*Chapter 5, Mastering Blueprints*, shows you how to master the creation of blueprints. An intimate deep dive into successful creation of Kickstart, SCCM, WIM, and Cloned blueprints.

*Chapter 6, Creating Approval Policies*, in this chapter, we will demonstrate step-by-step the creation and implementation of approval policies.

*Chapter 7, Installing and Configuring vRealize Automation Application Services 6.2*, how to integrate services such as LAMP stacks and SQL to the vRealize Automation user portal.

*Chapter 8, REST API and vRealize Orchestrator*, defines REST and covers the creation of workflows with the vRealize Orchestrator, as well as integrating the workflows with vRealize Automation.

*Chapter 9, Integrating vRealize Operations Manager*, provides an overview of vRealize Operations Manager 6.0, as well as exploring the benefits of integrating it with vRealize Automation.

*Chapter 10, Customizing the End User Portal Experience*, teaches you how to customize the end user portal, including services, catalog items, icons, entitlements, and business groups.

*Chapter 11, Troubleshooting vRealize Automation 6.2*, gives you information on some of the common errors and pitfalls that occur during installation and maintenance, as well as how to resolve them.

*Chapter 12, References for vRealize Automation 6.2*, gives you information on where to go to get deeper knowledge about certain topics and self-help resources.

## What you need for this book

In this book, we use two physical servers that are capable of running VMware ESXi 5.5 or later, and the following software:

- 1 x vRealize Automation 6.2 virtual appliance
- 1 x Identity Server virtual appliance
- 1 x Windows 2012 R2 Server to install the vRA IaaS components
- 1 x Windows 2012 R2 Server with SQL 2012 SP 2 installed
- 1 x DHCP Server
- 1 x E-mail Server (We will be using Exchange 2013 in our examples)
- 1 x vCenter 5.5 or later Server with SSO configured
- 1 x Windows Domain Controller
- 1 x ISO and VM for Windows and Linux

You can also use any reasonably powerful computer that runs Windows 2012 R2, Red Hat Linux 6, or later.

## Who this book is for

Mastering vRealize Automation 6.2 is intended for those who already have good working knowledge of the principles of vCenter and want to utilize automation and orchestration to provision their infrastructure as a service.

## Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "You can find these logs at `/var/log/vcac` in the vRA Appliance."



A block of code is set as follows:



```
cmd /c mkdir C:\Windows\Temp
cmd /c copy "\\192.168.1.2\e$\DeploymentShare\Applications\VMware
vcac Agent\7za.exe" C:\Windows\Temp
cmd /c copy "\\192.168.1.2\e$\DeploymentShare\Applications\VMware
vcac Agent\GugentZip_x64.zip" C:\Windows\Temp
```

Any command-line input or output is written as follows:

```
genisoimage -r -T -J -V "RHEL6AMD64" -b isolinux/isolinux.bin -c  
isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -o /  
localdirectory/Kickstart.iso /localdirectory/
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "There is an **Actions** tab in the **Blueprints** section."

 Warnings or important notes appear in a box like this. 

 Tips and tricks appear like this. 

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book – what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from [http://www.packtpub.com/sites/default/files/downloads/3779EN\\_ColoredImages.pdf](http://www.packtpub.com/sites/default/files/downloads/3779EN_ColoredImages.pdf).

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

## Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at [questions@packtpub.com](mailto:questions@packtpub.com) if you are having a problem with any aspect of the book, and we will do our best to address it.



# 1

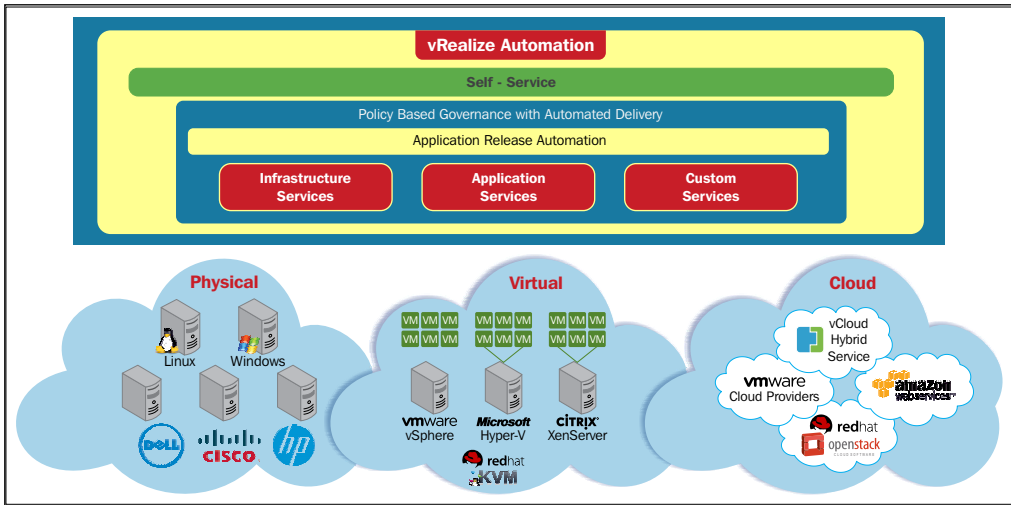
## Introducing vRealize Automation 6.2

In the first chapter, we are going to start by briefly explaining vRealize Automation 6.2, as well as cover some of the necessary fundamentals that play a part of the vRealize Automation universe. This includes defining **Infrastructure as a Service (IaaS)**, **Everything as a Service (XaaS)**, as well as the different editions of vRealize Automation that are available.

### Introduction to vRealize Automation

At its most basic definition, **vRealize Automation** (known as **vRA**, pronounced as "vee are ā") is a self-service web portal that provides XaaS. Not only in the strict XaaS definition, but *everything as a service* as a catch all includes IaaS, **Database as a Service (DBaaS)**, **IT as a Service (ITaaS)**, among other emerging services. vRA includes policy-based governance, and automatic delivery of user requests through a single web portal. It provides flexibility by allowing users to provision multi-vendor infrastructure and applications. vRA also provides business agility by automating processes that users normally depend (and wait) on their system administrators to take care of for them. End users no longer have to wait for their IT team to spin up *physical* or *virtual* servers, and install appropriate applications for them, along with all the hardening and compliance that a business demands. This can now be performed by the user with the orchestration of different workflows, all conducted by you, the IT professional.

The following figure shows an overview of vRA:



## Benefits of vRA

Agility and versatility are two of vRA's greatest benefits. Business users and developers can spin up and consume services, applications, and infrastructure that is needed to suit business demands. This leads to a faster life cycle of product development. Developers can design, test, and deploy new software much faster than using traditional methods, which includes an IT department, manually (or semi-automate) the deployment of foundational technologies needed in the product life cycle. Also, after a product has been deployed, developers can use the same self-service process to create an infrastructure for testing bugs in their code and resolving issues in the product as bug reports are filed by the consumer.

VMware's vRealize Automation provides a wealth of versatility. So much, in fact, it is quite an overwhelming product. In this book, we will delve into some of the more interesting points of vRA, such as workflow orchestration and integration with existing systems that may already be in your environment. These include:

- System Center Configuration Manager (SCCM)
- Kickstart
- vRealize Operations

Governance is one of the most difficult aspects of running a large, publicly traded company. **Sarbanes-Oxley (SOX)** and **Payment Card Industry (PCI)** audits are lengthy and take away hours from maintaining the foundational technologies necessary to keep a business running. This task would be a lot easier if you had a way to show an auditor a virtual paper trail of workflows and logs, detailing each event that occurs in your environment. Out of the box, vRealize Automation maintains detailed audit logs of what each user does while operating in the web portal. For example, user requests within the portal are logged under the **Items** tab (We will explore this more in *Chapter 4, Installing and Configuring vRealize Automation 6.2*). These items cannot be deleted without manually deleting tables out of the SQL database that controls the vRA IaaS services. Additionally, e-mails are sent to the users' designated manager for approval, if you configure vRA to do so. Also, we can tightly control our software and server licenses by limiting how many of these resources can be deployed in our environment.

Operational costs when it comes to compute, software, and power are often difficult to track. What if we could associate a per day cost of running a **Red Hat Enterprise Linux (RHEL)** virtual machine? Also, what if your business department has a fixed budget for your project or a limitation to what can be spent on infrastructure during a 30-day cycle? We can easily assign costs to all the aspects of the infrastructure from storage, networking, CPU, and memory, as well as applications. This gives us strict control over the amount of compute we provide to our customers or coworkers.

## Beginning of vRealize Automation 6.2

vRealize Automation 6.2 is the latest version of VMware's self-service web portal that is used to provide IaaS and XaaS to users on demand. The genesis of this product began in 2005, when a global financial services company called Credit Suisse had a need to roll out their virtualization strategy, which was comprised of thousands of virtual machines. Since they are a private and investment bank, they would be subject to all types of compliances, regulations, and governance.

After a few years of using the software they developed to maintain the wealth of virtual machines in their infrastructure, they formed a separate company named **DynamicOps**.

Launched in January 2008, DynamicOps was founded by Leslie Muller, who led the charge in developing what was then called *DynamicOps Cloud Suite*. The suite included DynamicOps Cloud Automation Center, which provided the foundation of what vRA is today. VMware bought DynamicOps in 2012 and shortly after renamed the product **vCloud Automation Center (vCAC)**.




vCAC was developed through major versions, such as 4, 5, and 6. Each version featured significant code changes under the hood, as VMware developers begin to remove the legacy DynamicOps code and make it their own. They also began to integrate with multiple vendors in an effort to offer everything as a service (XaaS).

In the fall of 2014, vCAC was rebranded again to the version we know today, vRealize Automation.

Why would VMware rebrand an already well-know product? According to their FAQ, *Realize* describes the power their management software gives customers to gain insights, make decisions, and take action across the entire IT landscape.

vRealize Automation 6.2 is now integrated with **vRealize Operations (vROps)**. It was formally known as **vCenter Operations (vCOps)**. You can read about the former version of vROps from *Packt vCenter Operations Manager 5.6.2*.

[  Integration means you will be able to view the health as well as operational trends through a single pane of glass, the vRA portal. ]

## Available editions of vRealize Automation 6.2

At the time of writing, vRA comes in three different flavors: Standard, Advanced, and Enterprise. Since the Standard edition is not available as a stand-alone product, we will not cover it beyond the next chapter. However, it is available as part of the vRealize Suite, which includes vRealize Operations Manager 6, among other products. For the examples used in this book, we will be focusing on the vRealize Automation 6.2 Enterprise edition. This means some of the examples used here may not be for your version. The most up-to-date information can be found at <http://www.vmware.com/products/vrealize-automation/compare>.

The editions are explained in more detail in the sections coming later. However, take a look at these charts for a detailed comparison of the three different editions of vRA:

Features	Standard	Advanced	Enterprise
<b>vRealize Automation</b>			
VMware infrastructure services, cloning only, vRO integration	Yes	Yes	Yes
Multi-vendor, multi-cloud infrastructure, and multi-vendor SW provisioning		Yes	Yes

Features	Standard	Advanced	Enterprise
Custom services (XaaS), approvals, reclamation, chargeback, multi-tenancy		Yes	Yes
Application services			Yes
<b>Services Provisioned and Managed</b>			
Import existing virtual machines	Yes	Yes	Yes
Infrastructure services (vSphere and vCloud only)	Yes	Yes	Yes
Day 2 operations for infrastructure: reconfigure and snapshot	Yes	Yes	Yes
Infrastructure services (multi-vendor virtual, physical, and public Cloud)		Yes	Yes
Custom services		Yes	Yes
Application services (virtual, public, and public Cloud)			Yes
Day 2 operations for applications: update, rollback, scale-in, and scale-out			Yes
<b>Software Deployment Mechanisms</b>			
Hypervisor and vApp cloning	Yes	Yes	Yes
Multi-vendor software deployment tools (Puppet, Chef, Kickstart, SaltStack, and so on)		Yes	Yes
Deploy integrated multi-tier applications			Yes
Leverage existing services in new application deployments			Yes
<b>Governance and Controls</b>			
Business rule, resource allocation, and infrastructure service definition policies	Yes	Yes	Yes
Multi-tenancy and approvals		Yes	Yes
Application definition			Yes
<b>Business Management</b>			
Chargeback and cost display throughout the product		Yes	Yes
Integration with VMware vRealize Business Standard Edition		Yes	Yes
<b>Solution Extensibility</b>			
vRealize Orchestra Integration	Yes	Yes	Yes
Optional vRealize Automation Development Kit (SDK)	Yes	Yes	Yes

Features	Standard	Advanced	Enterprise
VMware Cloud Management Marketplace solutions	Yes	Yes	Yes
Advanced Service Designer		Yes	Yes

## vRealize Automation Standard edition

This is the entry point into vRA. It supports a singular level of infrastructure services. This means it is limited to supporting vSphere and vCloud only. The Standard edition includes support for snapshots and **vRealize Orchestrator (vRO)**. (It was formally referred to as **vCenter Orchestrator (vCO)**). Also, it only supports server and **Virtual Appliance (vApp)** cloning to deploy new infrastructure.

Since the Standard edition is the entry point, you do not get any of the business management components, such as charge back capabilities and integration with the vRealize Business Standard edition, which is an IT cost management program. You also do not get multi-tenancy support and approvals, and all of the multi-vendor software deployment tools, such as **Windows Image Management (WIM)**, **Kickstart**, and **System Center Configuration Manager (SCCM)**.

## vRealize Automation Advanced edition

The Advanced edition includes the limited features of the Standard edition, but also adds a wealth of additional features, making it more appropriate for larger IT landscapes.

Multi-vendor, Cloud, and software provisioning are included, as well as the ability to provision and manage physical, virtual, and public Cloud infrastructures. This is also the first edition that introduces the multi-vendor software deployment tools we eluded to earlier in the chapter.

We also get multi-tenancy and approvals, as well as the chargeback and the vRealize Business integration.

Finally, you get access to the **Advanced Service Designer (ASD)**, which allows vRA to use an external vRO server and add the vRO plugins as endpoints.

## vRealize Automation Enterprise edition

Last, but not least is the most powerful edition of vRA. With the Enterprise edition, you get all of the features included in the Advanced edition. Additionally, you get gems, such as integration with *Puppet*, *Chef* and *SaltStack*. When it comes to compliance and governance, you get the added ability to create application definitions and release automation policies. The Enterprise edition allows you to deploy integrated multitier applications and leverage existing services in new application deployments.

Also, there is a big feature difference between Advanced and Enterprise application services. Only the Enterprise edition gives you the capability of provisioning applications across virtual, private, and public Cloud spaces.

## Summary

In this chapter, we defined vRealize Automation 6.2. We discussed the benefits of vRA and its associated features. We also discussed a bit of the history behind the product.

The final sections covered the three different editions: vRA Standard, vRA Advanced, and vRA Enterprise, as well as what's included in each edition.

In the next chapter we are going to dive into the technology a little deeper and look at how some of the technology features included with vRA fit together and how they work.



# 2

## vRealize Automation Architecture and Feature Overview

In the previous chapter, we defined vRealize Automation 6.2 and discussed some of the high-level benefits and the variance between the three editions. We also mentioned that vRA is available as part of the vRealize Suite, which also contains vROps Manager. In this chapter, we are going to cover the features and the different components that comprise vRA. We will illustrate some of the high-level features and go into the details of some of the more important milestones included in the 6.2 release.

We will be covering the following topics in the chapter:

- The core components of vRealize Automation 6.2
- An overview of its high-level architecture
- The features of vRealize Automation 6.2
- Other features that have been improved in this release

## What edition should I choose?

As we mentioned in the previous chapter, there are three versions of vRA available for your infrastructure: Standard, Advanced, and Enterprise. Since the Standard edition is only available as part of the vRealize Standard Suite, it is merely mentioned as a footnote in this book. At the time of this writing, the **Manufacturer's Suggested Retail Price (MSRP)** of the Advanced edition was \$6,750.00, while the Enterprise edition was \$9,950.00. Pricing is on a per CPU basis. It should also be mentioned that there is no 60-day evaluation available with either of these products. This is likely because there is no way to begin to crack the code of its potential with vRealize Automation in a mere 60 days. Although you can download the software from VMware, you would have to contact a VMware authorized reseller to get an evaluation key. If you are interested in evaluating the product first, I recommend you sign up for VMware's **Hands-on Labs (HOL)**. It is an excellent resource and there are several vRealize Automation labs available.

If you previously purchased a vCAC license during the 6.0 or 6.1 series, you will be able to upgrade to vRA 6.2 at no additional cost. VMware only charges for the major version changes of their software, such as moving from vCAC 5.0 to 6.0.

The ultimate choice between the three vRealize Automation versions depends on your infrastructure, as well as the business challenges you are trying to address. The Standard and Advanced edition have limited capability. So much so that they will not be mentioned beyond this chapter. If your desire is to automate your environment across physical, virtual, and multi-vendor platforms, I encourage you to choose the Enterprise edition.

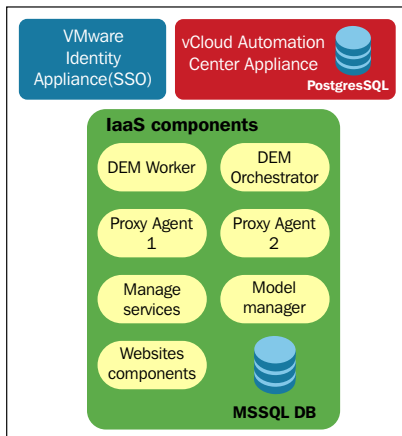
## vRealize Automation core components

VMware has made great strides to improve the core components of vRealize Automation. In the 5.2 version of vCAC, all the software components had to be installed on Windows servers. Starting with the 6.0 version, this has been reduced to two vApps and one Windows server.



In addition to the three core components, you must have vCenter 5.x or later, a domain controller, an e-mail server, and a SQL server to get the most basic functionalities out of vRealize Automation. It is recommended by VMware that you have a dedicated database server to host the SQL databases in a production environment.

Before we look at each core component in detail, let's have a high-level view of the vRA components using the following figure:



Now, let's look at the components that deliver the functionality and how they work.

## vRealize Automation Identity Appliance

This is a SUSE Linux-based vApp that provides **Single Sign-On (SSO)** authentication for your environment's end users via **Lightweight Directory Access Protocol (LDAP)** or **Active Directory (AD)**. You could use the SSO component included with vCenter 5.5.0b or later, but until the SSO code base is unified across all the VMware products, it is a best practice to use the vRealize Automation Identity Appliance.

## vRealize Automation Appliance

This is the vRealize vApp that serves as the web portal for end users. It is also based on SUSE Linux and features a Postgres database. You can deploy multiple vRealize Automation Appliances for high availability.

## IaaS server

The IaaS server is a Windows server that acts as the engine under the hood of the end user web portal. There are over 30 prerequisites involved in configuring the IaaS server, which is mainly composed of server roles and features, as well as **Microsoft Distributed Transaction Coordinator (MSDTC)**, a SQL database, and batch service rights. We will introduce a script that helps speed up this process in the next chapter. The IaaS server features several components that warrant deeper explanation, which will be explained in the following sections.



- **Model manager**

The model manager is the IaaS server component that manages vRealize Automation's data repository. It is also responsible for vRealize Automation's **Representational State Transfer (REST)**ful API. The model manager is divided into two components: the model manager web service and the model manager data component. Please note that these two components have to be co-located. However, this does not mean you have to install the SQL database on the IaaS server.

- **DEM components**

On an IaaS server, you will have at least have one Worker **DEM** (short for **Distributed Execution Manager**), and one Orchestrator DEM. You may have more of each, depending on your environment and deployment. For example, you may have multiple of each for redundancy purposes. DEMs call the model manager to see if there is any work that needs to be performed. If there is work, DEMs pull the work item and process the job. In the event of multiple DEMs, they can work together; they handle the workload and provide the desired redundancy level.

A Worker DEM performs the jobs requested in the vRealize Automation user portal. It also pulls any workflows and scripts necessary to complete the job. If the workflow or script associated with the job request has been used previously, the Worker DEM will grab that from the model manager's local cache.

An Orchestrator DEM monitors the Worker DEM's preprocessing workflows and schedules them for execution.

- **Agents**

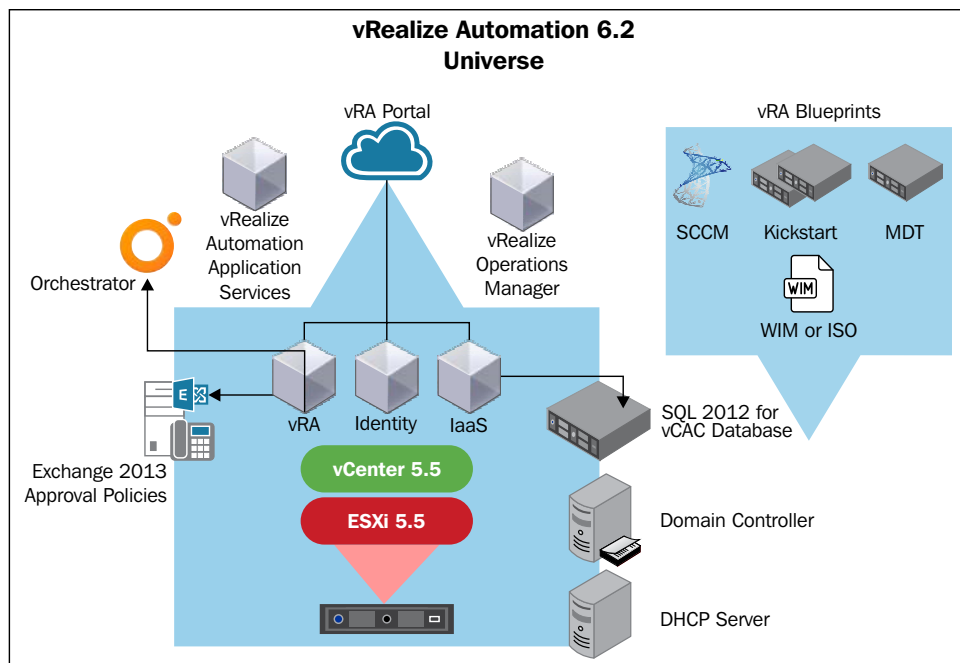
vRealize Automation uses agents to integrate with external systems. For example, if you want to deploy **virtual machines (VMs)** in your existing vCenter infrastructure through vRA, you will need to install a vSphere agent within it. There are a number of agents available for installation, depending on your version of vRA. These agents are stored on the IaaS server, but are installed and configured through the web portal. There are agents for Citrix XenServer, and Microsoft Hyper-V. There are also agents for **Windows Management Instrumentation (WMI)** and **Virtual Desktop Infrastructure (VDI)**, for example Horizon View.

There are also agents which are specifically used for Windows and Linux. These allow you to run scripts after the machine is provisioned, as in the case of Sysprep, or adding the user requesting the machine to be added to the local admin group.

We will go into more details about the architecture when we start laying out our **Proof Of Concept (POC)** in *Chapter 3, Designing and Building a vRealize Automation 6.2 Infrastructure*.

## High-level architecture overview

Now that we have an understanding of the core components of a vRealize Automation 6.2 infrastructure, let's look at how these components fit into a basic environmental topology using the following figure:



The preceding diagram displays all of the additional components that make up vRA. The three core components are in the middle, and the additional components that will be necessary for you to unleash the power of vRealize Automation surround it.

We will discuss how to tie in server components, such as **Dynamic Host Configuration Protocol (DHCP)**, Exchange, and SCCM in the next chapter.

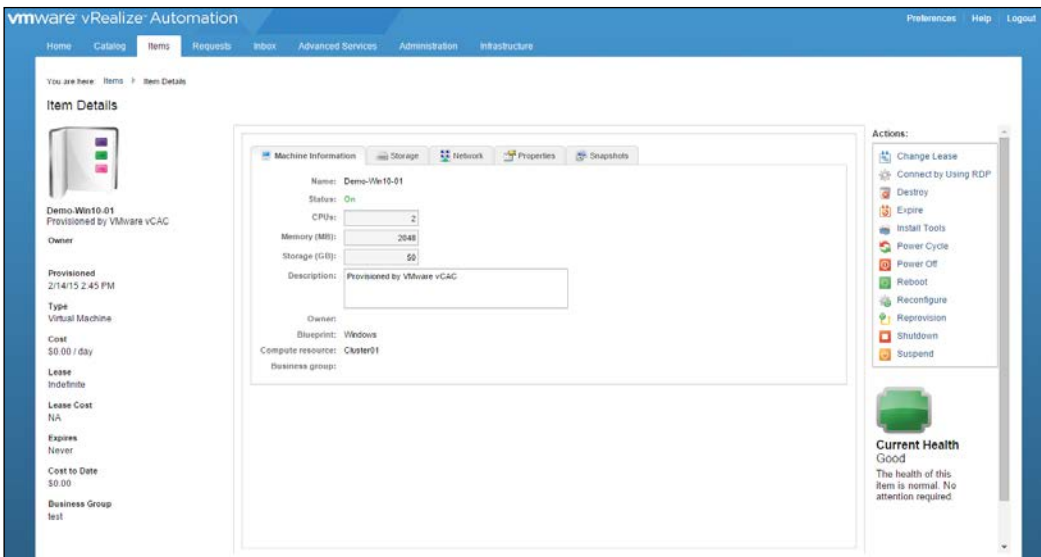
## Features of vRealize Automation 6.2

vRealize Automation 6.2 is the latest version of VMware's self-service web portal that provides IaaS, XaaS, and other *as a service* features to users on demand. This version now is tightly integrated with vROps (formerly known as vCOPS). In addition to supporting more multi-vendor platforms, vRA now has operational intelligence and analytics, along with the health badges and monitoring features you may have experienced with vCOPS (*VMware vCenter Operations Manager Essentials*, Lauren Malhoit, available here: <https://www.packtpub.com/virtualization-and-cloud/vmware-vcenter-operations-manager-essentials>).

Additionally, vRA allows you to integrate with VMware's **Namespace Extension (NSX)**, which is a software defined network solution.

Here are some of the exciting features of vRA 6.2. We will explain some of the features in more detail in the later sections:

- vROps integration is used to view the health of VMs in the environment as shown in the following screenshot:



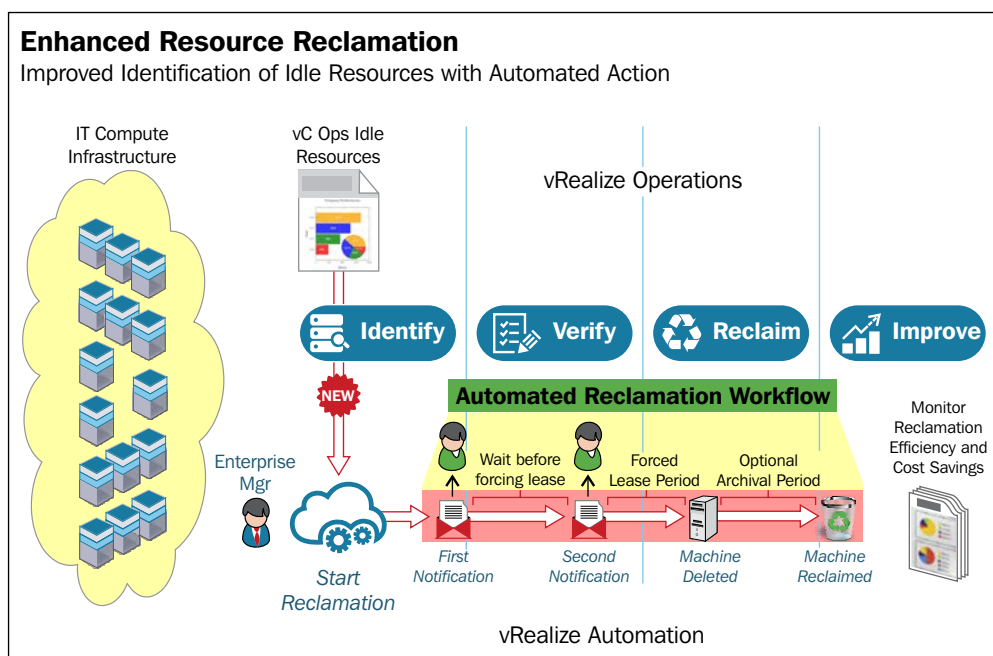
- vROps can be used by vRealize Automation to identify idle VMs based on configurable criteria

- Unused resources from the VM can now be reclaimed
- Enhanced support for vSphere 6, XenDesktop 7, and vCloud Air
- One-click generation of support bundles across all vRealize Automation components
- Pay-as-you-go support
- CloudClient
- Simplified UI scripting

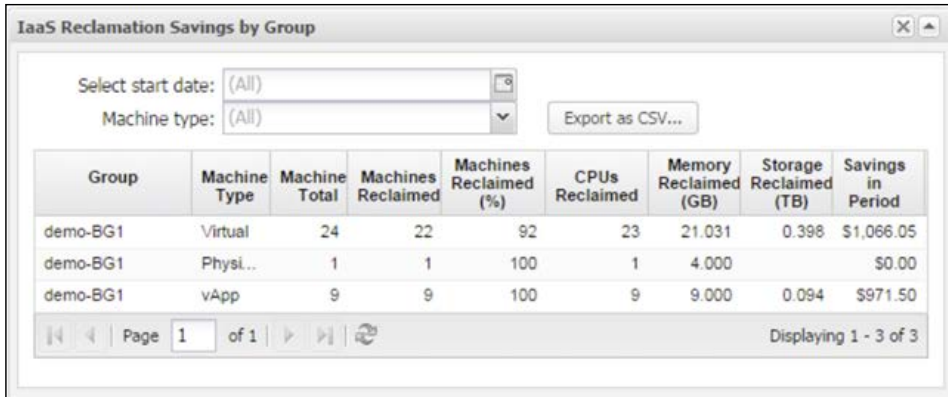
## Integration with vROps

One of the most exciting features of vRealize Automation 6.2 is its integration with vRealize Operations. If you have previous experience with the formerly named vCenter Operations Manager, you know how powerful this tool is as a standalone product. With regard to what we have discussed so far about vRA, imagine being able to view the health badges and trending analysis of your VMs through the vRA portal. It is the definition of convergence.

Since vRealize Operations will give us an insight into the amount of compute resources being consumed by VMs, we can now leverage this information with built-in workflows to allow us to reclaim unused sources. Take a look at the following chart for a high-level view of reclaiming an idle VM:



Since we can reclaim idle VMs and compute resources, we can show direct savings back to the business units that may have provisioned the infrastructure. For example, if we assign costs to our compute resources, we can present a report showing the amount of savings. The following screenshot demonstrates an example:



The screenshot shows a window titled "IaaS Reclamation Savings by Group". It includes a "Select start date:" dropdown set to "(All)", a "Machine type:" dropdown set to "(All)", and an "Export as CSV..." button. Below is a table with the following data:

Group	Machine Type	Machine Total	Machines Reclaimed	Machines Reclaimed (%)	CPUs Reclaimed	Memory Reclaimed (GB)	Storage Reclaimed (TB)	Savings in Period
demo-BG1	Virtual	24	22	92	23	21.031	0.398	\$1,066.05
demo-BG1	Physi...	1	1	100	1	4.000		\$0.00
demo-BG1	vApp	9	9	100	9	9.000	0.094	\$971.50

At the bottom, there is a pagination control showing "Page 1 of 1" and "Displaying 1 - 3 of 3".



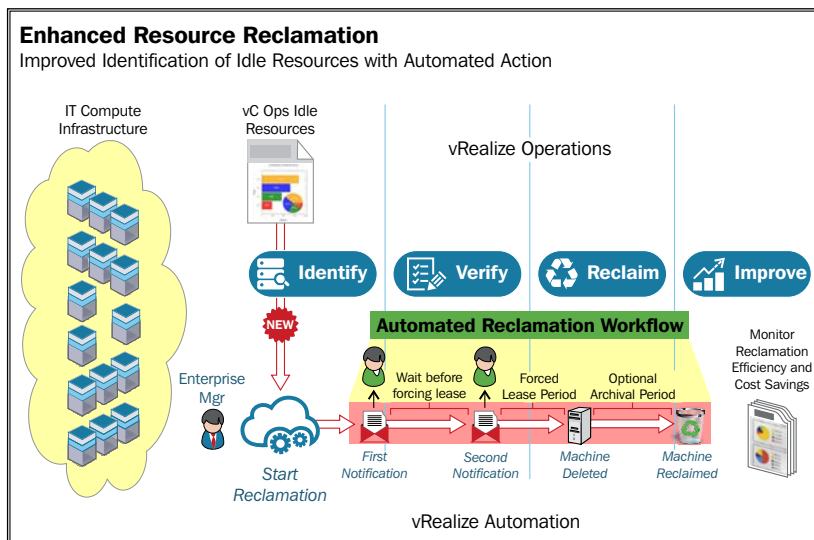
Please remember that you must already have vROps 6 installed in order to use the above features. It is also available as part of the vRealize Suite, but is not bundled with vRealize Automation 6.2.

## CloudClient

CloudClient is a command-line utility, which provides verb-based access across the vRealize Automation APIs, such as IaaS, applications, and vRO. It was developed as a separate product and was released in the fourth quarter of 2014. However, it is now integrated into vRealize Automation 6.2. It provides the following features:

- It focuses on providing an easy-to-use command-line interface for the IT administrator, where scripting and CLI use is more feasible than being involved in direct API calls
- It provides a stable interface, while underlying APIs may change over time
- It provides common security; exception handling; JSON, CSV and tabular formatting; file export; auto login for scripting (such as passwords and key files); and auto generated documentation

Take a look at the following screenshot for details:



## Pay-as-you-go support

In vRealize Automation 6.2, we have the ability to create cost profiles, which are typically relevant for small and medium sized businesses. VMware created this feature to work specifically with their vCloud Air OnDemand service. A great article about the details of this feature, as it continues to evolve, can be found here: <http://blogs.vmware.com/vcloud/2015/03/connecting-vrealize-automation-vcloud-air.html>

## Other features improved in this release

The recent benefits alone would be compelling enough to implement vRealize Automation 6.2. However, the foundation of vRealize Automation was established in previous versions and refined in this release. Out-of-the-box workflows, such as linked clones, WIM, SCCM, and Kickstart allow you to leverage multi-vendor solutions from a single web portal. vRO ties in and allows you to create customized workflows to tailor vRA to your unique infrastructure needs. Let's look at some of these features in more detail.



Please remember the features being discussed from this point forward pertain to the Enterprise edition of vRA 6.2. However, there may be portions that pertain to the Advanced edition as well.

## Blueprints

Blueprints are instruction sets to build vApps, VMs, or servers in general. This can be to a multitude of endpoints, including vSphere, Hyper-V, XenServer, or physical servers. Blueprints are one of the first items to setup in vRA and is necessary before you can deploy anything. You can define custom properties and policies to streamline deployments from blueprints. All of the options of blueprints will be covered in depth in the next chapter.

## Build profiles

Build profiles are a great way to store custom properties to use with blueprints. For example, you can build a profile called Windows 2012 R2 VMs. Within this build profile, you can define things, such as MAC addresses, whether a CD-ROM should be attached or not, network settings, the type of operating system, and whether snapshots should be allowed. Once this is created, you can attach it to multiple Windows-based blueprints. It saves you time by preventing the need to insert redundant custom properties across multiple blueprints.

## Clone deployment

Clone deployment is a great feature and is available across all the three editions of vRA 6.2. With this feature, you can create identical VMs based on snapshots or templates that you created in vCenter. A great example of its usage in an infrastructure would be to create a **Linux, Apache, MySQL, and PHP (LAMP)** stack. This is a common set up for web developers. As a vRA administrator, you can create a LAMP template or a snapshot of an existing LAMP VM, and let users deploy additional VMs based on the template or snapshot. This is considered a basic setup and will be covered briefly in the next chapter.

## WIM deployment

WIM deployment is a great feature if you do not have **System Center Configuration Manager (SCCM)** and want to leverage customized Windows images created through **Windows Deployment Services (WDS)** or **Microsoft Deployment Toolkit (MDT)**. You have to configure vRA to interact with your DHCP server and mount a customized VMware ISO during the deployment process to get this to work. This will be explained in detail in *Chapter 3, Designing and Building a vRealize Automation 6.2 Infrastructure*.

## SCCM deployment

If you are familiar with the power of SCCM and already have it in your environment, you can leverage it to provide servers through vRealize Automation. Users can login and request a Windows 2012 R2 server, for instance. vRA 6.2 will send a call to SCCM to deploy the Windows operating system, whether it is a server or desktop, and physical or virtual, depending on your blueprints.

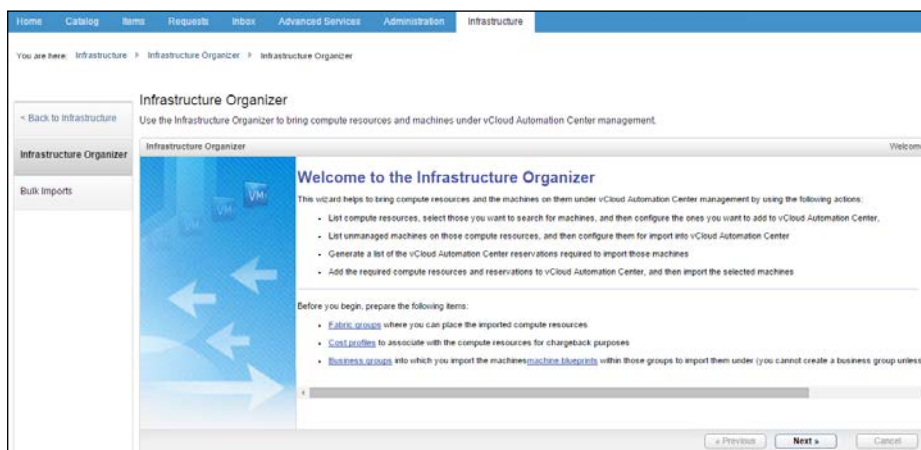
## Kickstart deployment

Kickstart is a great way to automate the deployment of Linux-based operating systems, including VMware's ESXi software. Most heterogeneous enterprise environments have a Kickstart server set up and deploying Linux. There is no need to reinvent the wheel. With vRA 6.2, we have built-in workflows to tie into our existing Kickstart server and provision infrastructure to the users who request it through the vRA web portal.

## Infrastructure Organizer

The Infrastructure Organizer is a component of the vRA 6.2 web portal and is a wizard used to import compute resources and VMs into the vRealize Automation fold.

Before you use it, you have to have fabric and business groups, as well as blueprints established. Here is an example of when you could use it: Let's say you have a group of development or test virtual servers. Once development and testing concludes as part of the life cycle of product development, you may want to destroy the VMs and return the resources back to the infrastructure. You can use the **Infrastructure Organizer** (as shown in the following screenshot) to import the development and test servers into vRealize Automation, and assign lease policies to have the servers expire and be destroyed at the end of the development and testing lifecycle:







You would likely want to refrain from using this feature to import virtual infrastructure servers, such as Exchange, SQL, and domain controllers. These are best kept outside of the vRealize Automation purview.

## Associating costs to resources

You can assign costs to compute resources within the vRA 6.2 web portal. Under the **Infrastructure** tab, there is a **Compute Resources** setting. Within this section is **Cost Profiles**. This is where you can assign a cost per CPU, GB of memory, and/or GB per storage consumed. This is not required to be set up. However, it is nice to present these costs back to the business unit that consumes these resources.

## Logs for auditing and compliance

Each time a request for IaaS or XaaS is made, it is logged under the **Requests** tab. The end users can keep track of their own requests and administrators can view all the requests. These items are stored under the **Requests** tab in the web portal:

Request	Item	Description	Cost	Lease Cost	Status	Submitter	Submitted
43	RHEL7 Kickstart		Not Applicable	Not Applicable	Failed		11/7/14 10:59 AM
42	Windows 2012 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/7/14 10:18 AM
41	Windows 2012 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/6/14 4:05 PM
40	Windows 2012 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/6/14 3:55 PM
39	Windows 2012 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/6/14 3:51 PM
38	Windows 2008 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/6/14 3:34 PM
37	Windows 2008 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/6/14 3:26 PM
36	Windows 2012 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/6/14 11:50 AM
35	Windows 2012 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/6/14 11:43 AM
34	Windows 2012 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/6/14 11:19 AM
33	Windows 2012 R2 MDT Image Test		Not Applicable	Not Applicable	Failed		11/6/14 11:01 AM
32	Destroy - vRealize -Win-07		\$0.00 / day	Not Applicable	Successful		11/6/14 10:58 AM



You cannot remove requests, even if they are for testing. This is by design to keep track of requests for businesses that have strict governance and compliance regulations. The only way to remove these items is to delete them from the SQL database. It is recommended that you refrain from doing this without the guidance of a VMware support specialist.

## vRealize Orchestration (vRO)

vRealize Orchestration was known as vCenter Orchestration earlier and is used to simplify the automation of complex IT tasks. It integrates with vRA and we have dedicated the entirety of *Chapter 8, REST API and vRealize Orchestrator* to discuss it in detail. It is one of the most advanced features of the vRealize Automation stack and you will learn it faster if you already have a strong background in scripting and working with other orchestration products.

## Integration with VMware Horizon View desktops

Horizon View 6.2 is the latest version of VMware's VDI solution. If you use this in your environment, you can integrate it with vRealize Automation. A plugin is available from VMware's Solution Exchange at <https://solutionexchange.vmware.com/store> that ties in with vRO and allows administrators to run automated workflows in order to provide Horizon View desktops to end users.



For more information read *VMware Horizon View Essentials*, Peter von Oven, Packt Publishing.

## Tenants and SSO authentication

The architecture and features, no doubt, have you excited about vRealize Automation. So, how do you login and check it out? Well, first, we need to go over the Identity vApp and **Tenants** at a high level.

The vRealize Automation Identity Appliance serves as the SSO point, integrating with your LDAP or AD environment. However, we will be discussing (and referencing for the rest of the book) integration from an AD standpoint.

Once the Identity vApp has been installed and configured to integrate with AD, you will have to install the vRealize Automation vApp. This is a web-based portal that you and your users will login. You will use it to administer the environment and users will use it to request *anything as a service*. This is where you will set up Tenants. However, you will not be able to assign infrastructure administrators until you set up the IaaS server. For this reason, it is recommended that you get IaaS up and running first.

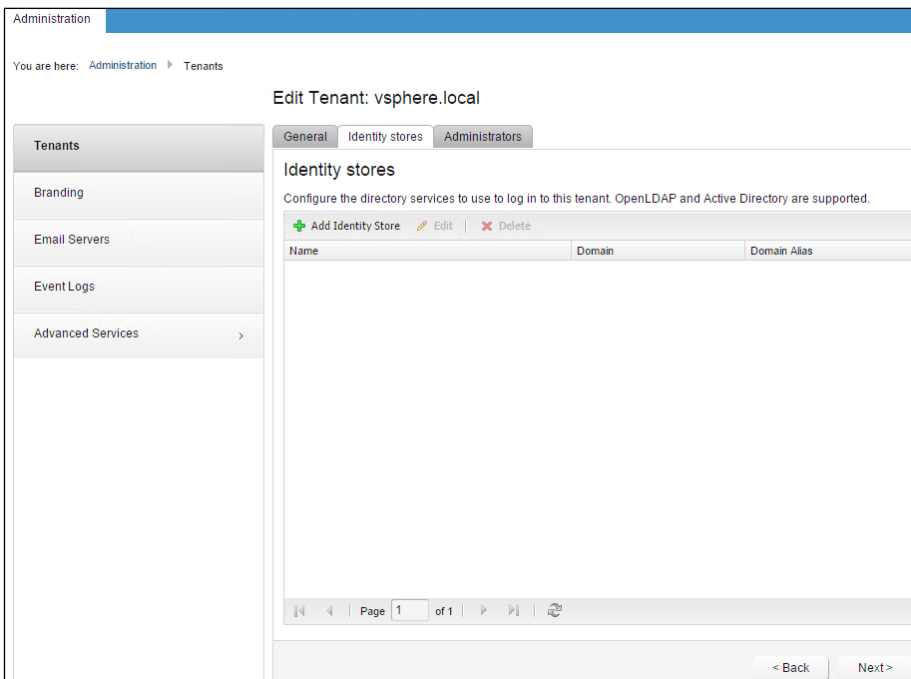
After the two vApps are set up and configured, you will have to install the IaaS software on a Windows 2012 R2 server (you can install this on a Windows 2008 R2 server too, but for the rest of the book, Windows 2012 R2 will be our default server).

There is a bit of prepping that you have to do before you install the IaaS software. You can download the IaaS software by pointing your web browser to the vRealize Automation vApp. If you run the software first, it will run a preinstall check against your Windows server. There are many roles and features that have to be installed. We will go into this with step-by-step instructions in the next chapter.

Once these three components are installed and configured correctly, you will need to configure Tenants.

To configure Tenants, you will login to the vRealize Automation web portal with the default SSO administrator account. Since vCenter 5.5 and newer, the default SSO administrator name is `administrator@vsphere.local`.

Tenants will already have **vsphere.local** configured by default. To configure the web portal so that you can login with AD credentials, we will create an **Identity Store**:



Once this is established, you can login with your AD account. However, you won't be able to do much until we go over the installation and configuration process in detail, which will be discussed in the next chapter.

## Summary

In this chapter, we detailed the architecture and features of vRA 6.2. We discussed some of the breakout new features in vRealize Automation, including its integration with vROps. We also covered older, but improved features, such as the ability to deploy from linked clones, WIM, SCCM, and Kickstart.

The final sections covered SSO and Tenants, which are needed to be able to login to the web portal and start unleashing the power of vRealize Automation.

In the next chapter, we are going to dive into the technology a little deeper and look at how some of the features included in vRealize Automation fit together and how they work.



# 3

## Designing and Building a vRealize Automation 6.2 Infrastructure

In the previous chapters, we have covered the basic principles necessary to understand vRealize Automation. Using what we have learned so far, we are going to start this chapter by taking a look at the design process and walking through the planning stages of a vRealize Automation 6.2 project. In this chapter, we will now start putting together a design and build vRealize Automation 6.2 from POC to production. With the knowledge gained from this chapter, you should feel comfortable installing and configuring vRA, which will be covered step by step in the next chapter.

In this chapter, we will be covering the following topics:

- Proving the technology
- Proof of Concept
- Proof of Technology
- Pilot
- Designing the vRealize Automation architecture
- Example of a design exercise

## **Proving the technology**

In this section, we are going to discuss how to approach a vRealize Automation 6.2 project. This is a necessary component in order to assure a successful project, and it is specifically necessary when we discuss vRA, due to the sheer amount of moving parts that comprise the software. We are going to focus on the end users, whether they are individuals or business units, such as your company's software development department. These are the people that will be using vRA to provide the speed and agility necessary to deliver results that drive the business and make money.

If we take this approach and treat our co-workers as customers, we can give them what they need to perform their jobs as opposed to what we perceive they need from an IT perspective. Designing our vRA deployment around the user and business requirements, first gives us a better plan to implement the backend infrastructure as well as the service offerings within the vRA web portal. This allows us to build a business case for vRealize Automation and will help determine which of the three editions will make sense to meet these needs.

Once we have our business case created, validated, and approved, we can start testing vRealize Automation. There are three common phases to a testing cycle:

- Proof of Concept
- Proof of Technology
- Pilot implementation

We will cover these phases in the following sections and explore whether you need them for your vRealize Automation 6.2 deployment.


## **Proof of Concept**

A POC is typically an abbreviated version of what you hope to achieve during production. It is normally spun up in a lab, using old hardware, with a limited number of test users.

Once your POC is set up, one of two things happen. First, nothing happens or it gets decommissioned. After all, it's just the IT department getting their hands dirty with new technology. This also happens when there is not a clear business driver, which provides a reason to have the technology in a production environment.

The second thing that could happen is that the technology is proven, and it moves into a pilot phase.

Of course, this is completely up to you. Perhaps, a demonstration of the technology will be enough, or testing some limited outcomes in VMware's HOL for vRealize Automation 6.2 will do the trick.

 Due to the number of components and features within vRA, it is strongly recommended that you create a POC, documenting the process along the way. This will give you a strong base if you take the project from POC to production.

## Proof of Technology

The object of a POT project is to determine whether the proposed solution or technology will integrate in your existing IT landscape and add value. This is the stage where it is important to document any technical issues you encounter in your individual environment. There is no need to involve pilot users in this process as it is specifically to validate the technical merits of the software.

## Pilot implementation

A **pilot** is a small scale and targeted roll out of the technology in a production environment. Its scope is limited, typically by a number of users and systems. This is to allow testing, so as to make sure the technology works as expected and designed. It also limits the business risk.

A pilot deployment in terms of vRA is also a way to gain feedback from the users who will ultimately use it on a regular basis. vRealize Automation 6.2 is a product that empowers the end users to provision everything as a service. How the users feel about the layout of the web portal, user experience, and automated feedback from the system directly impacts how well the product will work in a full-blown production scenario. This also gives you time to make any necessary modifications to the vRA environment before providing access to additional users.



When designing the pilot infrastructure, you should use the same hardware that is used during production. This includes ESXi hosts, storage, fiber or **Internet Small Computer System Interface (iSCSI)** connectivity, and vCenter versions. This will take into account any variances between platforms and configurations that could affect performance.

Even at this stage, design, attention to detail, and following VMware best practices is key. Often, pilot programs get rolled straight into production. Adhering to these concepts will put you on the right path to a successful deployment.

To get a better understanding, let's look at some of the design elements that should be considered:

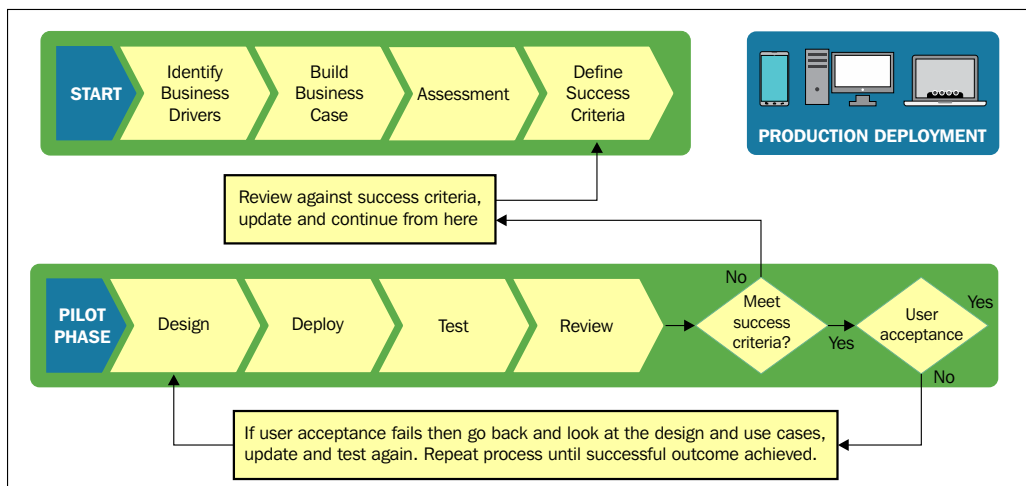
- **Size of the deployment:** A small deployment will support 10,000 managed machines, 500 catalog items, and 10 concurrent deployments.
- **Concurrent provisioning:** Only two concurrent provisions per endpoint are allowed by default. You may want to increase this limit to suit your requirements.
- **Hardware sizing:** This refers to the number of servers, the CPU, and the memory.
- **Scale:** This refers to whether there will be multiple Identity and vRealize Automation vApps.
- **Storage:** This refers to pools of storage from **Storage Area Network (SAN)** or **Network Attached Storage (NAS)** and tiers of storage for performance requirements.
- **Network:** This refers to LANs, load balancing, internal versus external access to web portals, and IP pools for use with the infrastructure provisioned through vRA.
- **Firewall:** This refers to knowing what ports need to be opened between the various components that make up vRA, as well as the other endpoint that may fall under vRA's purview.
- **Portal layout:** This refers to the items you want to provide to the end user and the manner in which you categorize them for future growth.
- **IT Business Management Suite Standard Edition:** If you are going to implement this product, it can scale up to 20,000 VMs across four vCenter servers.
- **Certificates:** Appliances can be self-signed, but it is recommended to use an internal **Certificate Authority** for vRA components and an externally signed certificate to use on the vRA web portal if it is going to be exposed to the public Internet.



VMware has published a Technical White Paper that covers all the details and considerations when deploying vRA. You can download the paper by visiting <http://www.vmware.com/files/pdf/products/vCloud/VMware-vCloud-Automation-Center-61-Reference-Architecture.pdf>.

VMware provides the following general recommendation when deploying vRealize Automation: keep all vRA components in the same time zone with their clocks synced. If you plan on using VMware IT Business Management Suite Standard Edition, deploy it in the same LAN as vCenter. You can deploy Worker DEMs and proxy agents over the WAN, but all other components should not go over the WAN, as to prevent performance degradation.

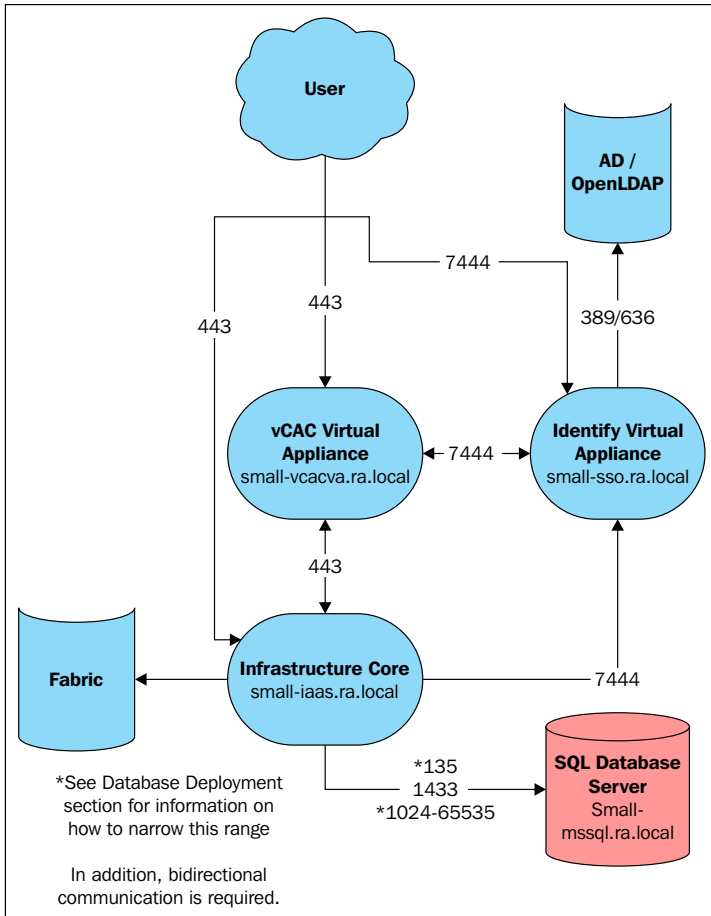
Here is a diagram of the pilot process:



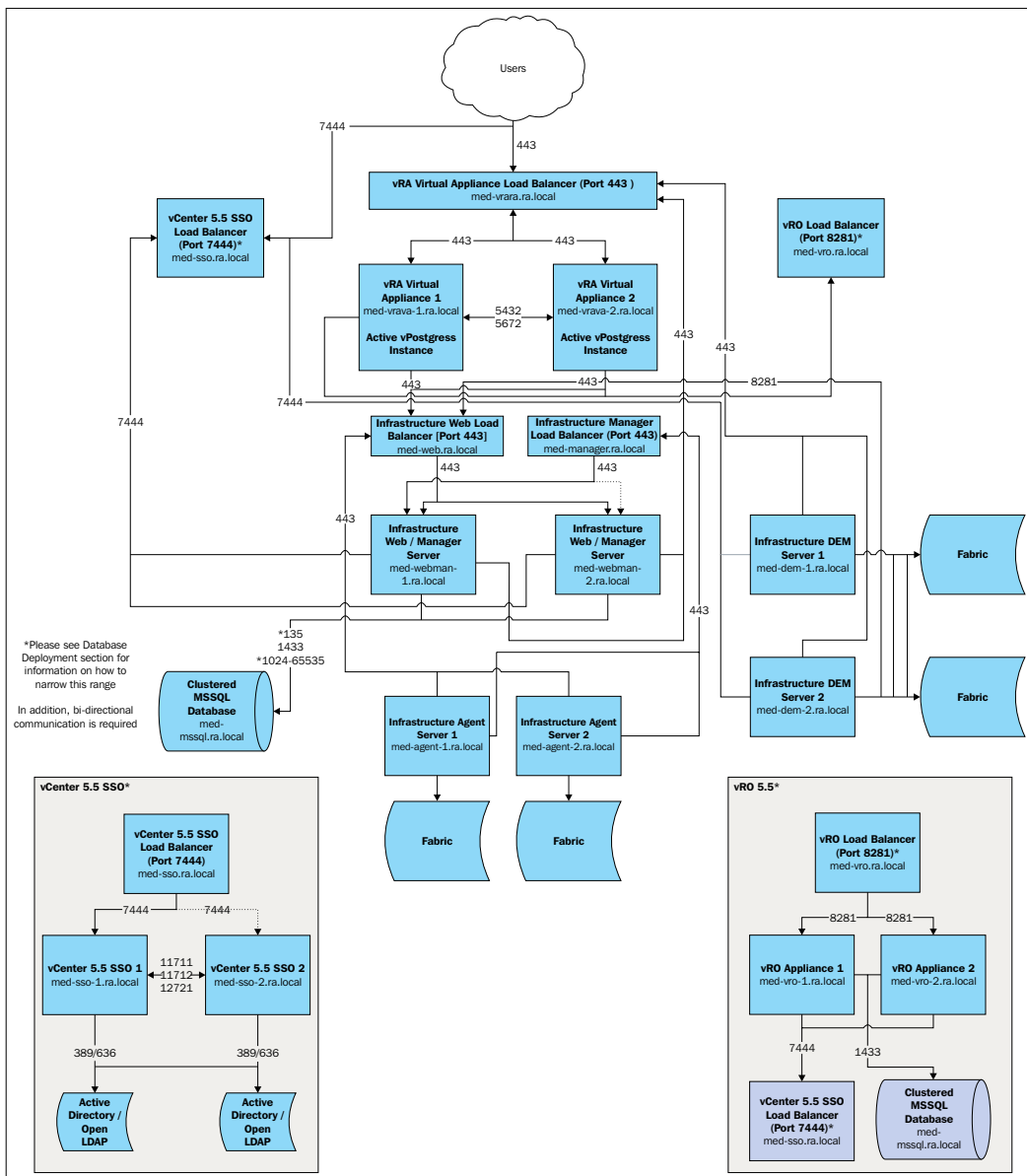
## Designing the vRealize Automation architecture

We have discussed the components that comprise vRealize Automation as well as some key design elements. Now, let's see some of the scenarios at a high level. Keep in mind that vRA is designed to manage tens of thousands of VMs in an infrastructure. Depending on your environment, you may never exceed the limitations of what VMware considers to be a small deployment.

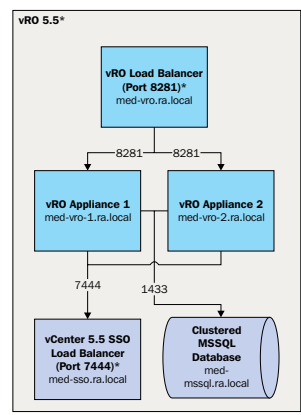
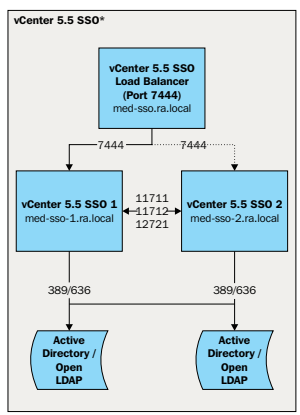
The following diagram displays the minimum footprint needed for small deployment architecture:



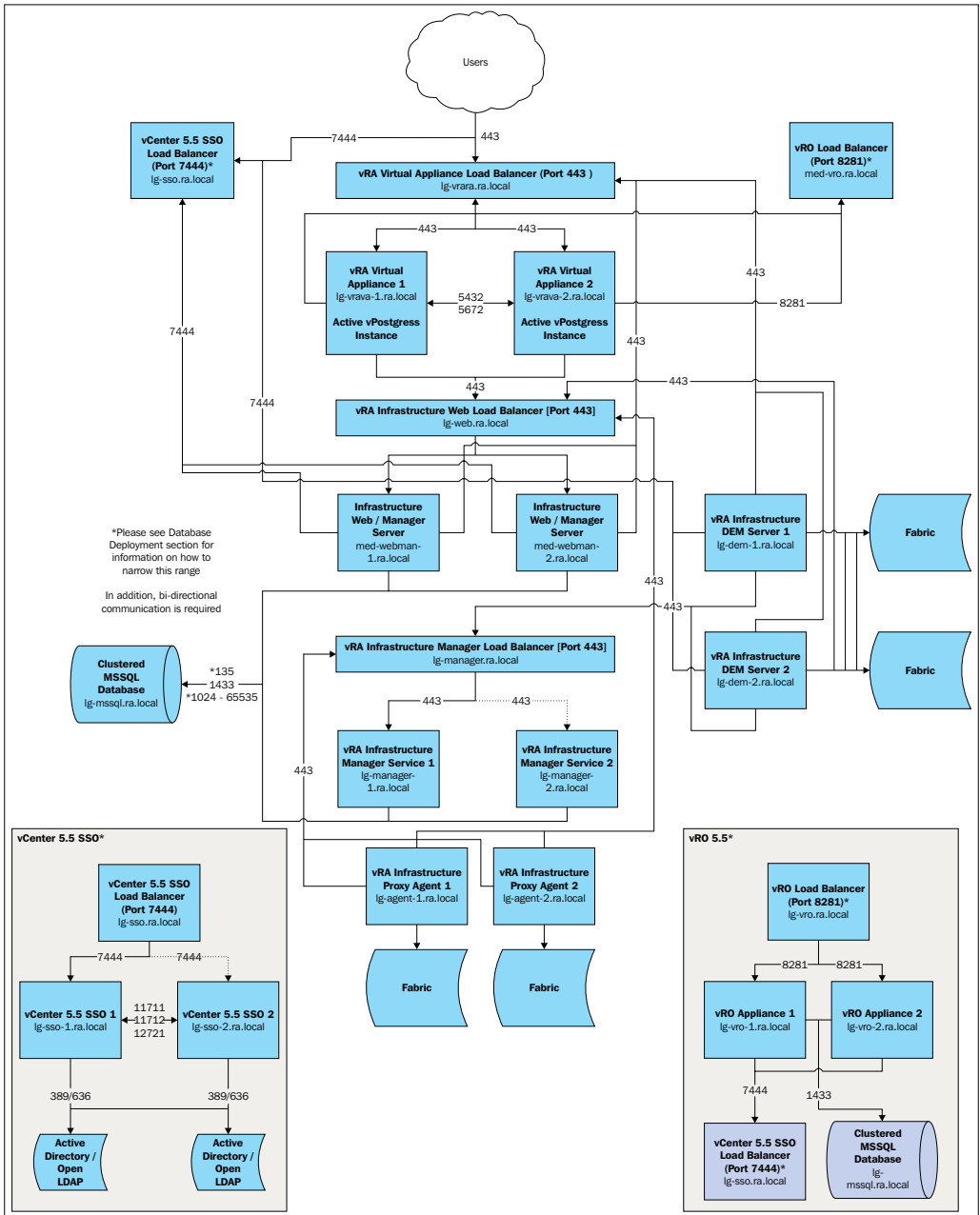
A medium deployment can support up to 30,000 managed machines, 1,000 catalog items, and 50 concurrent deployments. The following diagram shows you the minimum required footprint for a medium deployment:

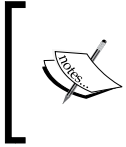


\*Please see Database Deployment section for information on how to narrow this range. In addition, bi-directional communication is required



Large deployments support 50,000 managed machines, 2,500 catalog items, and 100 concurrent deployments. The following diagram shows you the minimum required footprint for a large deployment:





For the purpose of this book, when we will install and configure vRA in the next chapter, we will implement a small deployment. However, this can easily scale to a medium or large deployment, depending on your environmental needs.

## Design considerations

Now that we understand the design elements for a small, medium, and large infrastructure, let's explore the components of vRA and build an example design, based on the small infrastructure requirements from VMware. Since there are so many options and components, we have broken them down into easily digestible components.

## Naming conventions

It is important to give some thought to naming conventions for different aspects of the vRA web portal. Your company has probably set a naming convention for servers and environments, and we will have to make sure items provisioned from vRA adhere to those standards. It is important to name the different components of vRealize Automation in a method that makes sense for what your end goal may be regarding what vRA will do. This is necessary because it is not easy (and in some cases not possible) to rename the elements of the vRA web portal once you have implemented them.

## Compute resources

**Compute resources** in terms of vRA refers to an object that represents a host, host cluster, virtual data center, or a public Cloud region, such as Amazon, where machines and applications can be provisioned. For example, compute resources can refer to vCenter, Hyper-V, or Amazon AWS. This list grows with each subsequent release of vRA.

## Business and Fabric groups

A **Business group** in the vRA web portal is a set of services and resources assigned to a set of users. Quite simply, it is a way to align a business department or unit with the resources it needs. For example, you may have a Business group named `Software Developers`, and you would want them to be able to provision SQL 2012 and 2014 on Windows 2012 R2 servers.

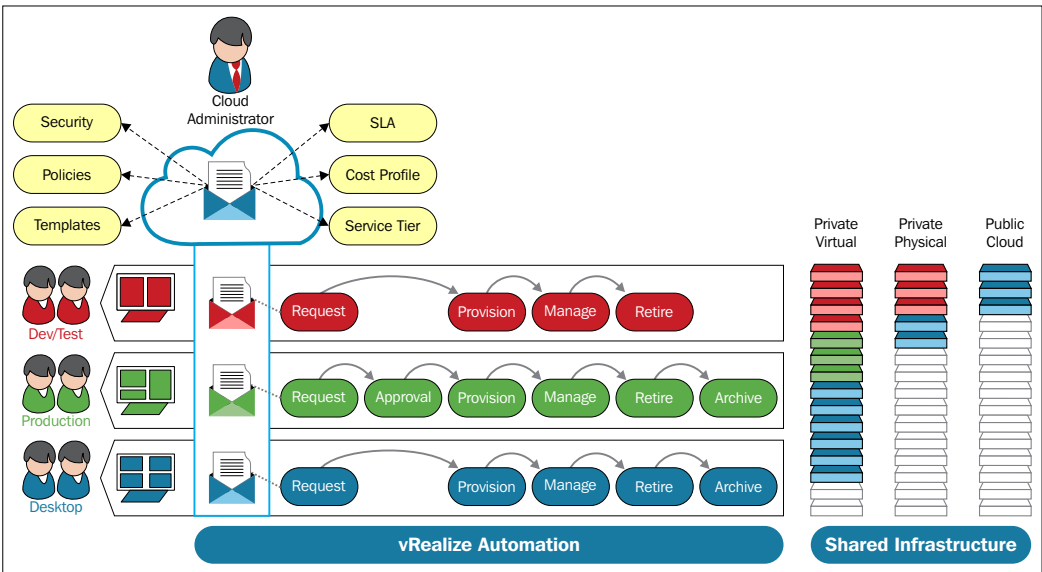
**Fabric groups** enable IT administrators to provide resources from your infrastructure. You can add users or groups to the Fabric group in order to manage the infrastructure resources you have assigned. For example, if you have a software development cluster in vCenter, you could create a Fabric group that contains the users responsible for the management of this cluster to oversee the cluster resources.

## Endpoints and credentials

Endpoints can represent anything from vCenter, to storage, physical servers, and public Cloud offerings, such as Amazon AWS. The platform address is defined with the endpoint (in terms of being accessed through a web browser) along with the credentials needed to manage them.

## Reservations

**Reservations** refer to how we provide a portion of our total infrastructure that is to be used for consumption by end users. It is a key design element in the vRealize Automation 6.2 infrastructure design. Each reservation created will need to define the disk, memory, networking, and priority. The lower their number, the higher will be the priority. This is to resolve conflicts in case there are multiple matching reservations. If the priorities of the multiple reservations are equal, vRA will choose a reservation in a round-robin style order:



---

In the preceding diagram, on the far right-hand side, we can see that we have **Shared Infrastructure** composed of **Private Physical** and **Private Virtual** space, as well as a portion of a **Public Cloud** offering. By creating different reservations, we can assure that there is enough infrastructure for the business, while providing a dedicated portion of the total infrastructure to our end users.

## Reservation policies

A **reservation policy** is a set of reservations that you can select from a blueprint to restrict provisioning only to specific reservations. Reservation policies are then attached to a reservation. An example of reservations policies can be taken when using them to create different storage policies. You can create a separate Bronze, Silver, and Gold policy to reflect the type of disk available on our SAN (such as SATA, SAS, and SDD).

## Network profiles

By default, vRA will assign an IP address from a DHCP server to all the machines it provisions. However, most production environments do not use DHCP for their servers. A **network profile** will need to be created to allocate and assign static IPs to these servers. Network profile options consist of external, private, **NAT** (short for **Network Address Translation**), and routed. For the scope of our examples, we will focus on the external option.

## Compute resources

Compute resources are tied in with Fabric groups, endpoints, storage reservation policies, and cost profiles. You must have these elements created before you can configure compute resources, although some components, such as storage and cost profiles, are optional. An example of a compute resource is a vCenter cluster. It is created automatically when you add an endpoint to the vRA web portal.


## Blueprints

We mentioned blueprints in the previous chapter, but need to provide more details about it to assist with the design process. Blueprints are instruction sets to build virtual, physical, and Cloud-based machines, as well vApps. Blueprints define a machine or a set of application properties, the way it is provisioned, and its policy and management settings. For an end user, a blueprint is listed as an item in the **Service Catalog** tab. The user can request the item, and vRA would use the blueprint to provision the user's request. Blueprints also provide a way to prompt the user making the request for additional items, such as more compute resources, application or machine names, as well as network information. Of course, this can be automated as




well and will probably be the preferred method in your environment.

Blueprints also contain workflow logic. vRealize Automation contains built-in workflows for cloning snapshots, Kickstart, ISO, SCCM, and WIM deployments. You can define a minimum and maximum for CPU, memory, and storage. This will give end users the option to customize their machines to match their individual needs.

 It is a best practice to define the minimum for servers with very low resources, such as 1 vCPU and 512 MB for memory. It is easy to **hot add** these resources if the end user needs more compute after an initial request. However, if you set the minimum resources too high in the blueprint, you cannot lower the value. You will have to create a new blueprint.

You can also define customized properties in the blueprints. For example, if you want to provide a VM with a defined MAC address or without a virtual CD-ROM attached, you can do so. VMware has published a detailed guide of the **Custom Properties** and their values. You can find it at <http://pubs.vmware.com/vra-62/topic/com.vmware.ICbase/PDF/vrealize-automation-62-custom-properties.pdf>.

 Custom Properties are case sensitive. It is recommended to test Custom Properties individually until you are comfortable using them. For example, a blueprint referencing an ISO workflow would fail if you have a Custom Property to remove the CD-ROM.

Finally, there is an **Actions** tab in the **Blueprints** section, but we will not be covering it in this book, as it is deprecated and will be removed in a future version of vRA.

## Users and groups

Users and groups are defined in the **Administration** section of the vRA web portal. This is where we would assign vRA specific roles to groups. It is worth mentioning when you login to the vRA web portal and click on users, it is blank. This is because of the sheer number of users that could be potentially allowed to access the portal and would slow the load time. In our examples, we will focus on users and groups from our Identity Appliance that ties in to Active Directory.

## Catalog management

Catalog management consists of services, catalog items, actions, and entitlement. We will discuss them in more detail in the following sections.

### Services

Services are another key design element and are defined by the vRA administrators to help group subsets of your environment. For example, you may have services defined for applications, where you would list items, such as SQL and Oracle databases. You could also create a service called **Operating Systems** where you would group catalog items, such as Linux and Windows. You can make these services active or inactive, and also define maintenance windows when catalog items under this category would be unavailable for provisioning.

### Catalog items

Catalog items are essentially links back to blueprints. These items are tied back to a service that you previously defined and helped shape the **Service Catalog** tab that the end user will use to provision machines and applications. Also, you will entitle users to use the catalog item.

### Entitlements

As mentioned previously, entitlements are how we link business users and groups to services, catalog items, and actions. You can create entitlements from users and groups, or link them to existing Business groups, as discussed in the previous chapter.

### Actions

Actions are a list of operations that gives a user the ability to perform certain tasks with services and catalog items. There are over 30 out of the box action items that come with vRA. This includes creating and destroying VMs, changing the lease time, as well as adding additional compute resources. You also have the option of creating custom actions as well.

## Approval policies

Approval policies are the sets of rules that govern the use of catalog items. They can be used in the pre or post configuration life cycle of an item. Let's say, as an example, we have a Red Hat Linux VM that a user can provision. We have set the minimum vCPU to 1, but have defined a maximum of 4. We would want to notify the user's manager and the IT team when a request to provision the VM exceeds the minimum vCPU we have defined.

We could create an approval policy to perform a pre-check to see if the user is requesting more than one vCPU. If the threshold is exceeded, an e-mail will be sent out to approve the additional vCPU resources. Until the notification is approved, the VM will not be provisioned.

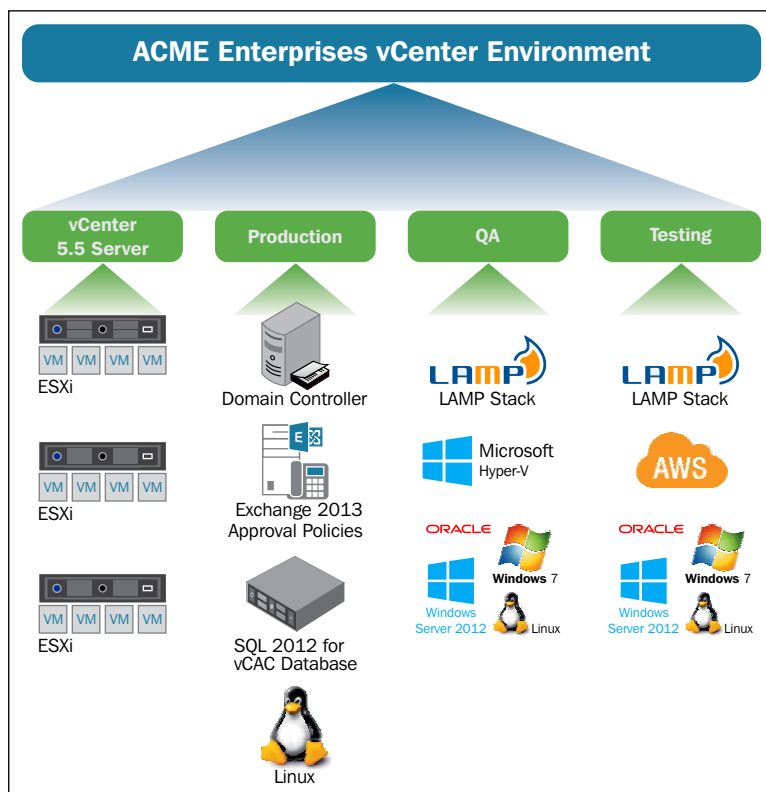
This subject will be covered in detail in *Chapter 6, Creating Approval Policies*.

## Advanced services

This subject will be covered in detail in *Chapter 8, REST API and vRealize Orchestrator*. However, it deserves a brief introduction so you won't be surprised when it is covered later. Advanced services is an area of the vRA web portal where we can tie in customized workflows from vRealize Orchestrator. For example, we may need to check for a file in the VM's operating system once it has been deployed. We need to do this to make sure that an application has been deployed successfully or a baseline compliance is in order. We can present vRealize Orchestrator workflows for end users to leverage in almost the same manner as we do IaaS.

## Example design exercise

We have covered a lot of different design considerations and core components in this chapter. Now that we have an understanding of small, medium, and large infrastructures, as well as the core components, let's put all the pieces together and build an example design. We will base it on the small infrastructure requirements from VMware. Since we have yet to go through the installation process, we will keep the design exercise limited to the concepts we have covered so far. For the purpose of the exercise, we will be designing a vRealize Automation 6.2 deployment for a fictitious company called ACME Enterprises. The company has vCenter 5.5 installed with a virtual cluster for testing, QA, development, and production. The following diagram shows what components each has in its environment presently:



As you can see from the preceding diagram, ACME Enterprises is a heterogeneous environment, with a mixture of operating systems, applications, virtualization technologies, desktops, and servers. We have done our due diligence and have the following requirements:

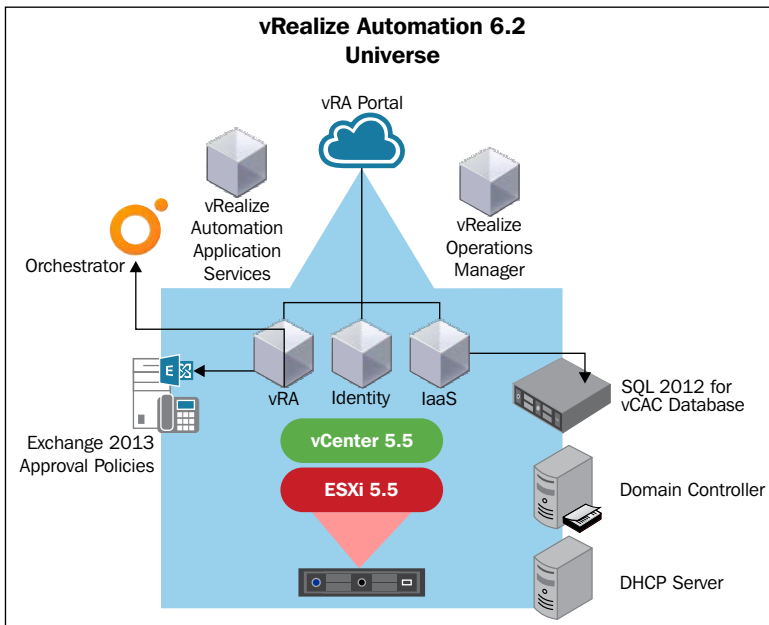
- We will implement vRA to the test environment as a phase I approach.
- We will give access to the vRA web portal to our IT team, as well as our software developers. Combined, this will consist of 10 IT team members and 90 software developers for a total of 100 users.
- The users will be allowed to provision Windows and Linux VMs in vCenter to accommodate their development needs. For baseline purposes, each VM will be allowed a minimum of 1 vCPU and 512 MB of RAM, and 50 G of storage. Our maximum will be 2 vCPUs, 1 GB of RAM, and 60 G of storage.

- Software developers will have to get approval from their managers as well as the IT team before their requested servers are provisioned. This serves two purposes: to bring awareness to management and to ensure the IT team can keep an eye on the compute resources being consumed in the test environment.
- VMs provisioned through vRA will be leased for a period of 30 days with the option to extend the lease after this period for 15 additional days. At the end of the lease, the VMs will be destroyed by vRA.
- The test environment has one subnet (192.168.1.x) and all the provisioned VMs will be on this subnet. DHCP will be used to provide the IP addresses since this is a test environment.

Since we have established the needs of ACME Enterprises, we can start to shape the vRealize Automation 6.2 design and calculate how much storage and compute will be needed to accommodate the rollout.

## vRealize Automation components

In this design exercise, we will need to install the Identity and vRA virtual appliances, as well as a Windows 2012 R2 Server for the IaaS roles, and leverage the existing SQL, DHCP, and Exchange server in the ACME Enterprises environment. We will use the compute and storage already present in the test environment cluster of vCenter:



---

## Calculating the amount of storage and compute resources

We learned earlier in the chapter about reservations and how they are portions of the total storage and compute available in the infrastructure. Now, we need to perform some calculations to make sure we can successfully design our vRA environment for ACME Enterprises in their test environment.

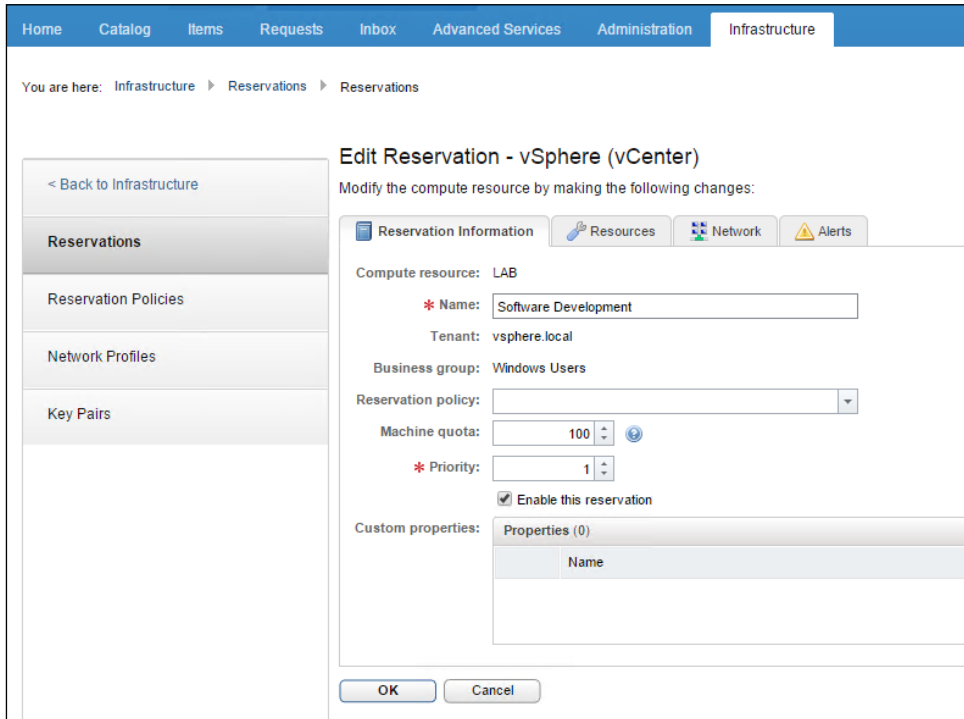
The best approach is to take the maximum compute resources that could be provisioned through vRA and add it together. Once you have the total, does it exceed the maximum amount of compute that can be served up in your vCenter environment? It is quite normal for users to consume everything you provide. Therefore, if you are going to allow them to consume four vCPUs as the maximum per VM, it is best to plan this out so that you have the capacity to support it in place. Remember we can control this utilization with approval policies, so as to ensure that users do not consume 100 percent of the portion of infrastructure we provide to vRA.

If we have a total of 100 users, as outlined in the previous section, do we want all 100 to be able to provision VMs? Do we want them to be able to provision as many as they want? It would be a bit of an embarrassment to implement vRealize Automation 6.2 and have the 100 user base consume the storage and compute instantly, with no room to grow.

We can control this pain point not only through approval policies, but also entitlements. Let's say out of the user base of 100, only 10 users can provision VMs and only 100 VMs can run in the vCenter test environment at one time. We can now make calculations to ensure we have enough resources to support our scenario.

With these items in mind, we would create a single reservation in the vRA web portal called `Software Development`, and allow them to have a total of 100 Gig of RAM out of the total amount available in the test environment cluster. CPU is not a configurable option in the **Reservations** tab's setting. However, we can specify that we want no more than 100 machines to be provisioned (**Unlimited** is the default value). Additionally, we can reserve the total amount of storage we need to fit our design example in or across multiple storage resources.

Take a look at the following screenshot as an example of this:



## Network considerations

Since DHCP is the default resource that vRealize Automation 6.2 uses to assign IP addresses, there is nothing additional that we would have to configure within the web portal. However, you would want to ensure that you have at least 100 IP addresses in the 192.168.1.x subnet on your DHCP server available to be assigned as the VMs are provisioned based on user demand.

## Naming the provisioned machines

vRealize Automation is very diverse and allows the provisioning user to name the machine, or you can also define **machine prefixes**. This is an automated way for vRA to provision machines with a naming convention previously defined. In this exercise, ACME Enterprises has a naming scheme of *Company Initials-Operating System-Environment-Number*. Therefore, based on the information provided for this exercise, we will create two machine prefixes, one for *AE-Win-Test-* and *AE-RH-Test-*. We will configure vRA to add sequential numbers to the end of the naming scheme, making sure to avoid any numbers that may already be in use in the test environment.

## Blueprints

We have discussed blueprints at this point and should now have a good understanding of their purpose. In this exercise, we will create two blueprints: one for Red Hat Linux and one for Windows 2012 R2. Be sure to name them something that will let the end user know who deploys them exactly what they will be getting out of them. For example, if the Windows 2012 R2 server already has SQL 2014 installed, we may want to call the blueprint *SQL 2014 on Windows 2012 R2*.

## Services and catalog items

In this exercise, we will create a single service named `Software Development` and two catalog items: `SQL 2014 on Windows 2012 R2` and `Red Hat Linux`. We will create entitlements to allow the ten predefined users out of our `Software Development` group at ACME Enterprises, the ability to provision, extend leases, and destroy provisioned VMs. The other 90 will have access to use the provisioned VMs through the portal, RDP, and SSH.

## Summary

In this chapter, we covered the design and build principles of vRealize Automation 6.2. We discussed how to prove the technology by performing due diligence checks with the business users and creating a case to implement a POC. We detailed considerations when rolling out vRA in a pilot program and showed you how to gauge its success.

We detailed the components that comprise the design and build of vRealize Automation, while introducing additional elements not covered in previous chapters. Lastly, we performed a design exercise to fit vRA in a test environment at ACME Enterprises, with the purpose of getting us to focus on design elements that will help in a smoother and fast setup once we install and configure vRealize Automation 6.2 in the next chapter.





# 4

## Installing and Configuring vRealize Automation 6.2

In this chapter, we are going to get vRealize Automation installed and configured in our environment. Based on the concepts covered in the previous chapter, we now have a roadmap that shows us how vRealize Automation works. Not only will we cover the step-by-step instructions to get vRealize Automation up and running, we will highlight some of the potential pitfalls that can be pain points when configuring the components. Before we perform the installation, we will cover some of the prerequisites as well as look at where to download the software. Once we have the software downloaded, we are going to focus on setting up the IaaS components of vRA, as it is the core of the product.

In this chapter, we will be covering the following topics:

- Preparing for the installation
- Installing the vRealize Automation Identity Appliance
- Installing the vRealize Automation Virtual Appliance
- Configuring the vRealize Automation Identity Appliance
- Configuring the vRealize Automation Appliance
- Installing and configuring the IaaS server
- Upgrading the vRealize Automation and Identity Appliances
- Configuring the vRealize Automation

## Preparing for the installation

Before we install and configure the software, let's define some of the prerequisites needed for the installation. It is strongly recommended that you perform these operations in a lab environment before rolling this into production. Keep the example design exercise we previously outlined in mind while performing the steps in this chapter.

## What you need to get started

For our examples in the rest of the book, we will be working with the following minimum requirements:

- 2 x ESXi hosts running the latest version of 5.5 or later.
- 1 x vRealize Automation 6.2 Virtual Appliance.
- 1 x Identity Server Virtual Appliance.
- 1 x Windows Server 2012 R2 to install the vRealize Automation IaaS components.
- 1 x Windows Server 2012 R2 with SQL 2012 SP 2 installed.
- 1 x DHCP server.
- 1 x E-mail server (we will be using Exchange in our examples).
- 1 x vCenter Server 5.5 or later with SSO configured (you will need to know the password of your `administrator@vsphere.local` account).
- 1 x Active Directory account for use as the vRealize Automation administrator. This account will be a member of the **Tenant** and **Infrastructure** administrator groups. For our example, we will use `vcs@domain.local`. Make sure this account is also an administrator in vCenter.
- 1 x Windows Domain Controller.
- 1 x ISO and VM for Windows and Linux.

The IaaS and SQL servers need to be joined to the domain.

These are the minimum components to get vRealize Automation 6.2 up and running. However, we will be covering advanced topics in the next chapter, such as SCCM and Kickstart deployments. You will need to have these servers set up in order to take advantage of those features.



Additionally, you should review the Release Notes published before and during the installation. This book references the vRealize Automation 6.2 notes listed at <https://www.vmware.com/support/vcac/doc/vrealize-automation-62-release-notes.html>.

## Downloading the software

In order to download the vRealize Automation 6.2 software, you must first go to [www.vmware.com](http://www.vmware.com), and login to the **My VMware** portal. If you do not have an account, you can sign up at no charge.

Once you're logged in, click on the **All Downloads** icon on the left-hand side of the portal, then click on the **All Products** tab. Under **Infrastructure and Operations Management**, search for **VMware vRealize Automation**. Click on **View Download Components**.

You will see the different versions available for download. We will download the Enterprise edition of vRealize Automation 6.2:

Product	Release Date	
Advanced		
VMware vRealize Automation 6.2.0	2014-12-09	<a href="#">Go to Downloads</a>
Enterprise		
VMware vRealize Automation 6.2.0	2014-12-09	<a href="#">Go to Downloads</a>
VMware vRealize Automation Application Services 6.2.0	2014-12-09	<a href="#">Go to Downloads</a>
Development Kit		
VMware vRealize Automation 6.2.0 - Development Kit	2014-12-09	<a href="#">Go to Downloads</a>

The Enterprise edition has several components to download. For the examples in this book, we will download the following components:

- vRealize Automation Appliance- OVA file
- vRealize Automation Identity Appliance for VMware vRealize Automation 6.2.0- OVA file

Download these two appliances to a shared folder in your lab environment.



The software you need to install on the Windows Server 2012 R2 you have dedicated as your IaaS resource is already part of the vRealize Automation Appliance and is not available as a separate download.

## Installing the vRealize Automation Identity Appliance

Follow these steps to install the vRealize Automation Identity Appliance:

1. Open vSphere and login to your lab's vCenter instance. Note that we will be using the traditional vSphere GUI instead of the web-based version.
2. Select **File** in the upper left-hand corner of vSphere and select **Deploy OVF Template**.
3. Browse to the location where you downloaded the software in the previous section and select the vRealize Identity OVA file. Then hit **Next**.
4. The next screen will list **Product Name**, **version**, **vendor**, and **Publisher**. Then hit **Next**.
5. Accept the **VMware License** on the next screen and hit **Next**.
6. Accept the default name and select the installation location. Then hit **Next**.
7. Select the cluster where you want to deploy the Identity Appliance and hit **Next**.
8. Select the appropriate resource pool and hit **Next**.
9. Select what storage location you would like to have the appliance installed in and hit **Next**.
10. Select the disk format. By default, it is selected for **Thick Provisioning**. However, you can change this to **Thin Provisioning** if you wish. Make your choice and hit **Next**.
11. In the **Network** section, select the appropriate network selection and hit **Next**. Remember, we will keep all the machines on the same subnet for the examples in this book.

12. In the **Properties** section, define the following:
  - Root password
  - SSH (enabling it is recommended)
  - Hostname
  - IP address
  - Gateway
  - Subnet
  - DNS
13. Once you've completed this, hit **Next**. Place a check mark in the box that says **Power On**, then hit **Finish**.
14. In the **DNS** section, make a static entry for your vRealize Automation Virtual Appliance. In this exercise, we will use `vRasso.domain.local` as the **fully qualified domain name (FQDN)**.

## Installing the vRealize Automation Virtual Appliance

Follow these steps to install vRealize Automation Virtual Appliance:

1. Open vSphere and login to your lab's vCenter instance. Note that we will be using the traditional vSphere GUI instead of the web-based version.
2. Select **File** in the upper left-hand corner of vSphere and select **Deploy OVF Template**.
3. Browse to the location where you downloaded the software in the previous section and select the `vRealize Appliance OVA` file. Then hit **Next**.
4. The next screen will list the **Product Name**, **version**, **vendor**, and **Publisher**. Then hit **Next**.
5. Accept the **VMware License** on the next screen and hit **Next**.
6. Accept the default name and select the install location. Then hit **Next**.
7. Select the cluster where you want to deploy the vRealize Automation Appliance and hit **Next**.
8. Select the appropriate resource pool and hit **Next**.

9. Select what storage location you would like to have the appliance installed in and hit **Next**.
10. Select the disk format. By default, it is selected for **Thick Provisioning**. However, you can change this to **Thin Provisioning** if you wish. Make your choice and hit **Next**.
11. Under the **Network** section, select the appropriate network selection and hit **Next**. Remember, we will keep all the machines on the same subnet for the examples in this book.
12. In the **Properties** section, define the following:
  - Root password
  - SSH (enabling it is recommended)
  - Hostname
  - IP address
  - Gateway
  - Subnet
  - DNS
13. Once you've completed this, hit **Next**. Place a check mark in the box that says **Power On**, then hit **Finish**.
14. In the **DNS** section, make a static entry for your vRealize Automation Virtual Appliance. In this exercise, we will use `vRA.domain.local` as the FQDN.

## Configuring the vRealize Automation Identity Appliance

You might think at this point that it would make sense to configure the IaaS server and SQL server. However, IaaS is dependent on having the Identity and vRealize Automation Appliances configured properly. Follow these steps to configure the vRealize Automation Identity Appliance:

1. Open vSphere and login to your lab's vCenter. Make sure your Identity Appliance is powered on if you missed doing so in the previous section.
2. Open a web browser on a machine that is on the same subnet as the Identity Appliance. This ensures there are no firewalls in place. In our example, this would be at `https://vRA.sso.domain.local:5480`.

3. Since this is the first time you are logging in, you will get a certificate warning. This is because it uses the certificate that comes with the default appliance and is not a trusted publisher. Continue until you get to the screen with a user and password prompt.
4. Log in as `root` with the password you selected in the previous section.
5. You will end up on the **Admin Settings** page. Select the **Time Settings** tab. Be sure to configure it to a time source. If your lab has access to the Internet, you could set your time source to **0.pool.org**. However, if your lab does not have Internet access, you will need to use the host's time. Save your settings.
6. Next, click the **System** tab, then select **Time Zone**. Set the appropriate time zone and select **Save Settings**.



It may seem odd that the first thing we are concerned about is to ensure that the time is correct. However, if there is a time lag between the Identity, vRealize Automation Appliance, and IaaS server, it can introduce many strange errors. Examples of such errors will be covered in *Chapter 11, Troubleshooting vRealize Automation 6.2*.

The main functionality of the Identity Appliance is to provide a web portal on the vRealize Automation server to interact with the Domain Controller. This is configured under the **SSO** tab.

7. Click on the subheader titled **Host Settings** and type the FQDN of the Identity server in the same way you created it in DNS. In our example, this is `vRASSO.domain.local`. Save your settings.
8. Next, click on the **SSL** tab and select **Generate Self Signed Certificate**. The common name will be filled in for you automatically and will match the FQDN you entered in step 7. The country code should be filled in automatically as well. If it's not, please use an appropriate two letter country code. For example, use **US** for United States. You will have to fill out **Organization** and **Organization Unit**. We will use `Domain` for the **Organization** and `IT` for the **Organization Unit**. Click on **Apply Settings**. This will take a couple of moments to complete. If this were a production box, you would want to generate a certificate from your internal certificate authority and select the **Import PEM encoded certificate**. It is a best practice to make the certificate at least 2048 bits in length. Use the self-signed certificate in lab environments only.



9. On the **Active Directory** tab, you can opt to join the Identity Appliance to the domain. Although this is not required, we do it for completeness. Fill in the FQDN of the domain, as well as the username and password of an account that is authorized to join systems to the Active Directory Domain. Click on the **Join AD Domain** button. Once this is successfully completed, it will display **Joined to domain DOMAIN.LOCAL**, based on our example.
10. Click on the **SSO domain** tab and in the **System Domain** field, type `vsphere.local`. This is not an example, this is literal. This ties the Identity Appliance to the SSO established in vCenter. As of version 5.5 of vCenter, `vsphere.local` is the default, built-in local SSO domain. Use the username `administrator@vsphere.local` and the password used to setup vCenter SSO. Click on **Apply**. This will take several minutes to complete.



If your SSO password contains any \$ symbols, the password will not work for SSO and will have to be reset. If you do not know the password for `administrator@vsphere.local`, you can reset it through this VMware Knowledge Base article-**Unlocking and resetting the VMware vCenter Single Sign-On administrator password (2034608)**, which can be found at: [http://kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayKC&externalId=2034608](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2034608).

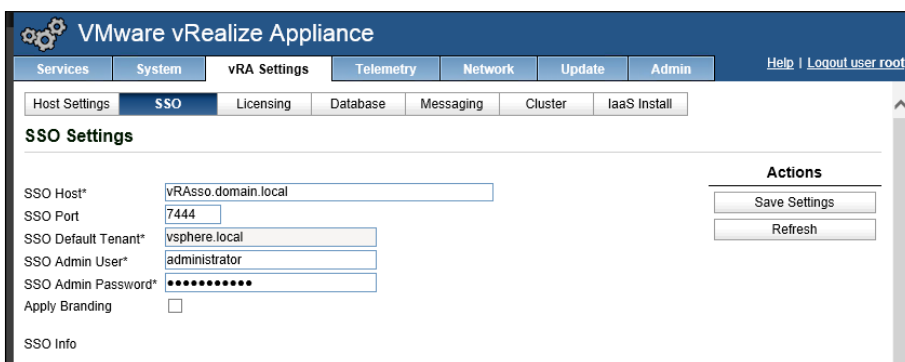
Congratulations. You have set up the first component of vRealize Automation 6.2. Proceed to the next section which details how to configure the vRealize Automation Appliance.

## Configuring the vRealize Automation Appliance

Following are the steps to configure the vRealize Automation Appliance:

1. Open vSphere and login to your lab's vCenter. Make sure your vRealize Automation Appliance is powered on if you missed doing so in the installation section.
2. Open a web browser on a machine that is on the same subnet as the vRealize Automation Appliance. This ensures there are no firewalls in place. In our example, this would be `https://vRA.domain.local:5480`.
3. Since this is the first time you are logging in, you will get a certificate warning. This is because it uses the certificate that comes with the default appliance and is not a trusted publisher. Continue until you get to the screen with a user and password prompt.

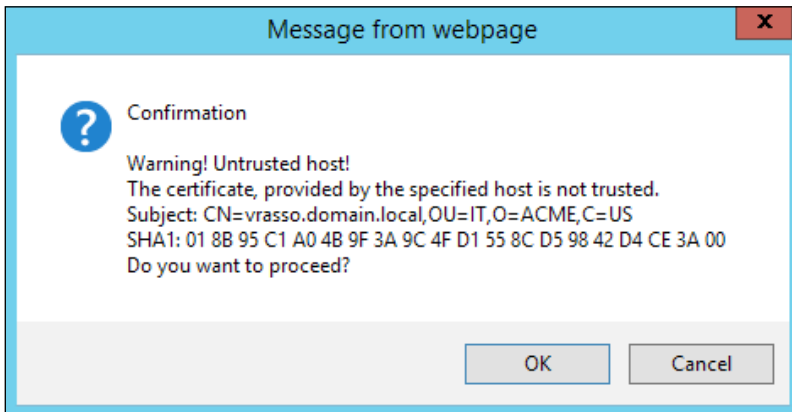
4. Login as `root` with the password you selected in the previous section.
5. You will end up on the **Admin Settings** page. Select the **Time Settings** tab. Be sure to configure it to a time source. If your lab has access to the Internet, you could set your time source to **0.pool.org**. However, if your lab does not have Internet access, you will need to use host's time. Save your settings.
6. Next, click on the **System** tab, then select **Time Zone**. Set the appropriate time zone and select **Save Settings**.
7. The main functionality of the vRealize Automation Appliance is configured under the **vRA Settings** tab. Click on it to continue.
8. Click on the subheader titled **Host Settings** and type the FQDN of the vRealize Automation Appliance as you created it in DNS. In our example, this is `vRA.domain.local`.
9. In the **SSL Configuration** section on the same page, select **Generate Certificate**. The common name should match the FQDN you entered in step 8. You will have to fill out the **Organization** and **Organization Unit**. We will use `Domain` for the **Organization** and `IT` for the **Organization Unit**. For the **Country Code**, please use an appropriate two letter country code. For example, use **US** for United States. Click on **Save Settings**. This will take a couple of moments to complete. If this were a production box, you would want to generate a certificate from your internal Certificate Authority and select the **Import PEM encoded Certificate**. It is a best practice to make the certificate at least 2048-bits in length. Use the self-signed certificate in lab environments only.
10. Click on the **SSO** tab and in the **SSO Host**, type `vRAsso.domain.local` (or whatever you have assigned to be the Identity Appliance in your environment). Also, type the password for **SSO Admin Password**. It is the password you used when setting up SSO in vCenter 5.5. Notice that the **SSO Port**, **SSO Default Tenant**, and **SSO Admin User** fields are prepopulated. Do not change these settings. Take a look at the following screenshot:





If your SSO password contains any \$ symbols, the password will not work for SSO and will have to be reset. If you do not know the password for administrator@vsphere.local, you can reset it through this VMware Knowledge Base article 2034608, which can be found at: [http://kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayKC&externalId=2034608](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2034608).

11. Once you've completed this, hit **Save Settings**. Since we are using a lab environment for the examples in the book, we will get a confirmation dialog box displayed, which asks for confirmation of the self-signed certificate from the Identity Appliance. Hit **OK**, as shown in the following screenshot:



12. Click on the **Licensing** tab and input the license. Remember that 60-day evaluation keys are not available in the same fashion as they are for most other VMware products. You will have to contact a VMware sales representative to get an evaluation key. Keep the key safe, as you will have to enter it a second time once you login to the vRealize Automation web portal. Click on **Submit Key** to validate your entry. Note that this is not instantly validated and can take up to 30 seconds to complete.
13. Click on the **Database** tab. It should be configured automatically and be in a **Connected** state.
14. Click on the **Messaging** tab. Again, this should be preconfigured and in a **Connected** state.

15. Finally, you can click on the **Cluster** tab and configure the vRealize Automation Appliance for high availability by standing up another instance of the vRealize Automation Appliance. However, for the purpose of this exercise, we will leave this option unconfigured. However, it is worth noting that you can collect the logs from all of the nodes in your setup from this tab.
16. We will discuss the **IaaS Install** tab later in this book. However, note that this is where you can download the IaaS software installer, as well as database scripts and the tools necessary to boot WIM images during the deployment process.

Congratulations! You have setup the second component of vRealize Automation 6.2. Proceed to the next section which details how to configure the IaaS server.

## Installing and configuring the IaaS server

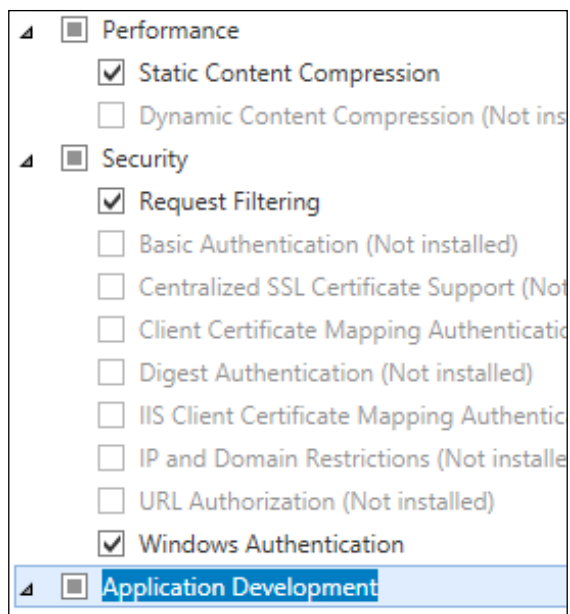
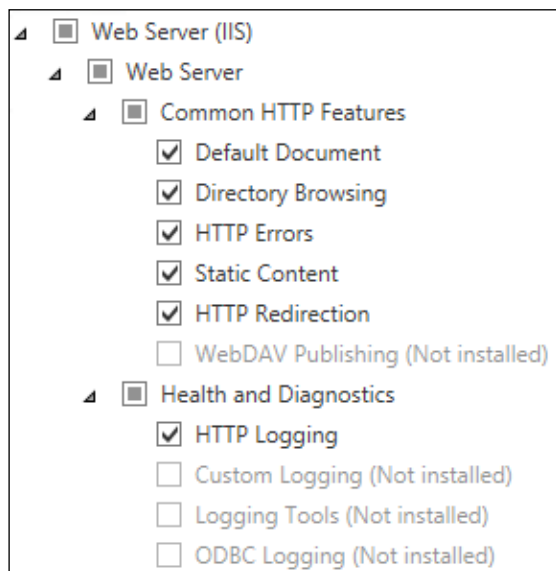
Login to the domain which joins the Windows Server 2012 R2 you created to serve as your IaaS Server. This server should also be on the same subnet as the vRealize Automation and Identity Appliances to ensure there is no interference with a firewall. For this exercise, the IaaS server's FQDN will be `vRAiaas.domain.local`.

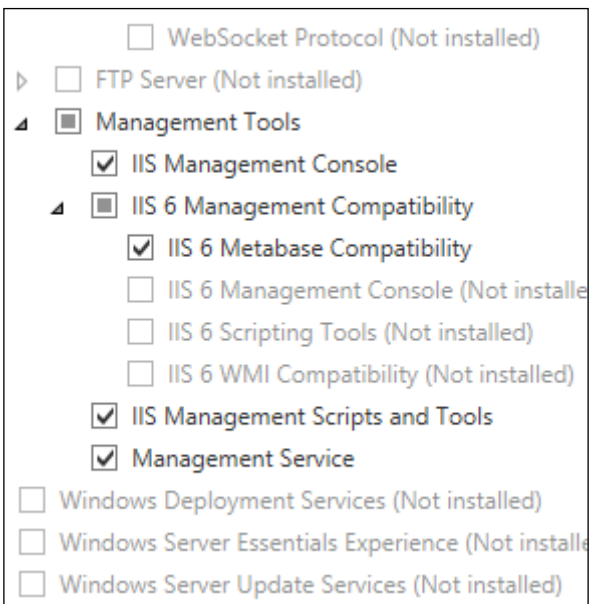
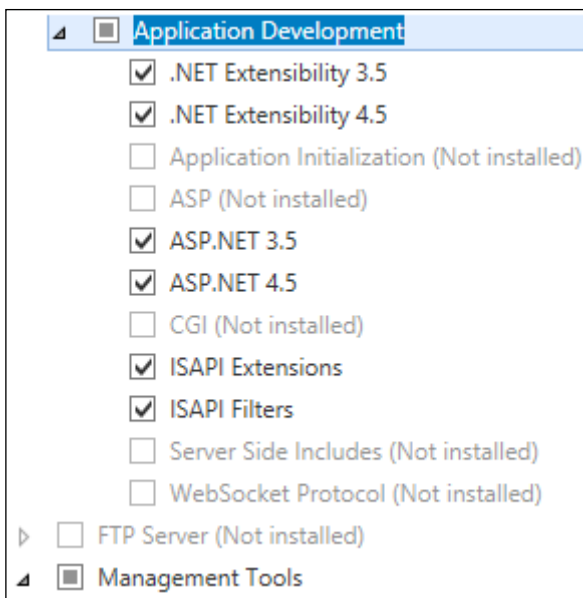
Before we go any further, it is important to highlight the roles, features, and prerequisites necessary to ensure a successful installation of the IaaS software. All the following activities outlined should be performed as a domain administrator. However, at the bottom of this section is the URL of a PowerShell script that will perform all of these steps for you. It is recommended that you use it.

You will need the following in place before downloading and installing the IaaS software:

- Java JRE version 7, update 51 (64-bit) is the minimum you need. Install it and then put `java.exe` using the following steps:
  1. Click on **Control Panel**, then on **System**.
  2. Click on the **Advanced System** settings, which is in the far left-hand corner.
  3. Click on the **Advanced** tab, and then on the **Environment Variables** button.
  4. In the **System Variables** section, click on the **New** button.
  5. In **Variable Name**, type `JAVA_HOME`.
  6. In **Variable Value**, type `c:\Program Files\Java\jre7`.
  7. Save your changes.

- Install the following web server roles:







Brian Graf (<http://www.vtagion.com>) has made a great IaaS Prerequisite PowerShell script. It is highly recommended that you use this to setup the Windows-based roles, features, and configurations, as they can be quite tricky and time consuming to perform manually. You can download the script from [https://github.com/vtagion/Scripts/blob/master/vRA 6.2 PreReq Automation Script.ps1](https://github.com/vtagion/Scripts/blob/master/vRA%206.2%20PreReq%20Automation%20Script.ps1). Don't forget to right-click on the **File**, select **Properties**, and click on the **Unblock** button before you run it.

Now that our prerequisite requirements are satisfied, we can go ahead and download the IaaS software. Open a web browser on the IaaS server and navigate to <https://vRA.domain.local:5480/installer>.

This will take you to the same page as the **IaaS Settings** tab in the vRealize Automation Appliance. Click on the top option, which is **Download the IaaS installer**. As the website states, do not rename this file after you download it. Take a look at the following screenshot:

**vmware**

## VMware vCloud Automation Center IaaS Installation

### IaaS Installation

- Download the [IaaS Installer](#) to install the IaaS components on Windows, or to upgrade the IaaS components from vCloud Automation Center 6.0.1 to 6.1.1. Do not rename the files.  
Build 2041200
- Download the [Migration Tool](#) to migrate from vCloud Automation Center 5.2.1 or later to 6.1.1.
- For manual database installation, download the [database installation scripts](#).
- Download the [database upgrade scripts](#) to upgrade the database from vCloud Automation Center 6.0.1 to 6.1.1. You must upgrade the database before upgrading any IaaS components.

Prerequisites for IaaS Installation:

- [NET Framework 4.5.1](#) (Not needed for Windows Server 2012 R2)

### Provisioning Utilities

- Windows guest agent files (32-bit) (64-bit)
- [Linux guest agent packages](#)
- [PE Builder utility](#)

### Administration and Development Tools

- [vCloud Automation Center Designer](#)

Now that the installer is downloaded, launch it using administrative privileges. This means right-click on the installer and select the **Run as Administrator** option. Depending on your **UAC** settings (**User Access Control**) you may get a pop-up confirmation box. Hit **Yes** and continue.

Now follow the following steps:

1. You will see a welcome screen. Hit **Next**.
2. Now, you will see the license agreement screen. Agree to the license and hit **Next** to continue.
3. The vRealize Automation Appliance FQDN and port number will be filled in automatically. You will need to enter the root and the root password to continue. Also, place a tick in the box beside **Accept Certificate**. Once you've completed this, hit **Next**.
4. Select **Complete Install**, then hit **Next**.
5. A built-in prerequisite checker will run. Thank goodness you were prepared and already have this part resolved. The results of the check are displayed. Often times, Windows firewalls are not turned on and this can cause the prerequisite check to fail. However, there is a **Bypass** button that will allow the IaaS software to believe that all the conditions are satisfied. Hit **Next**.



It must be stressed that successful completion of this section is an absolute must. Failure to pass the prerequisite checks successfully will cause all sorts of issues later in the configuration process. It is acceptable, however, to leave the Windows firewall turned off if you manage server security with another product in production.

6. The next section details the account to use to run IaaS services, a passphrase to generate the encryption key that protects the data at rest in the database, and the SQL configuration information. It is a best practice to use a service account in this section. The service account must have logon as **Batch Job Rights**. If it doesn't, a warning box will appear with instructions. Also, use Windows Authentication instead of a local SQL account.



Often times, using a SQL local account causes failures in the installation process. Make sure you have added the domain account you are using to SQL and have assigned the account dbo and sysadmin roles. You could run the IaaS installer as the SQL account you want to use. Note that the IaaS software has to install the database. It will fail at this step if the database already exists. Hit **Next**, as shown here:

The screenshot shows the 'vCloud Automation Center Configuration' window with the 'Server and Account Settings' tab selected. The window title bar includes the VMware logo and a close button. Below the title bar, the text reads: 'Install and configure all the components on the local machine. The database can be installed on the same machine or on a different machine.'

The 'Server Installation Information' section contains the following fields and text:

- Local Server:
- User name:
- Password:  Confirm:
- Passphrase:  Confirm:

Text below the password fields: 'All services will run under the account specified above.'

Text below the passphrase fields: 'The passphrase is used to generate the encryption key that protects data while at rest in the database. Memorize the passphrase or store it in a secure location. You reuse the passphrase across the IaaS deployment so that each component has the same encryption key. The passphrase is also required when you upgrade the installation.'

To the right of the 'Server Installation Information' section, a list of components to be installed is shown:

- Manager Service
- Model Manager Web Service
- Model Manager Data
- IaaS Administration Portal
- DEM Orchestrator
- DEM Worker
- Web API

The 'Microsoft SQL Server Database Installation Information' section contains the following fields:

- Server:  Database name:
- Use Windows authentication
- Username:
- Password:  Confirm:

At the bottom right of the window, there are three buttons: '< Back', 'Next >', and 'Cancel'.

7. The next section allows you to configure the DEM names and install the vSphere Agent. Take the defaults and hit **Next**:

The screenshot shows a configuration window titled "vCloud Automation Center Configuration" with a sub-header "Distributed Executions Managers And Proxy vSphere Agent". Below the sub-header is the instruction "Install agents to integrate with your provisioning infrastructure or other external systems." The window is divided into two main sections. The first section, "Distributed Execution Manager Details", contains two text input fields: "DEM Worker name:" with the value "DEM" and "DEM Orchestrator name:" with the value "DEO". The second section, "vSphere Agent Details", is preceded by a checked checkbox labeled "Install and configure vSphere agent". It contains two text input fields: "vSphere Agent name:" with the value "vSphereAgent" and "Endpoint name:" with the value "vCenter". At the bottom right of the window are three buttons: "< Back", "Next >", and "Cancel".

8. Enter the FQDN of the vRA Appliance. In our example, it is `vRA.domain.local`. In the **SSO tenant field**, hit the **Load** button. It will populate with `vsphere.local`. Next to **Certificate**, hit the **Download** button. Be sure to put a check in the **Accept Certificate** box.
9. In the **SSO Administrator** credentials, enter `administrator@vsphere.local` and the password. Be sure to hit the test link. If the test fails, make sure you do not have any strange characters, specifically a \$ symbol in your password. If you do, this portion will fail. Finally, enter the FQDN of the IaaS server (the one you are logged into now and performing these steps).

For our example, we've used `vRAiaaS.domain.local`. Hit **Next**:

The screenshot shows the 'vCloud Automation Center Configuration' window with the 'Component Registry' tab selected. The window title bar includes the VMware logo and a close button. The main content area is titled 'Component Registry' and contains the following fields and controls:

- Server:** A text input field containing 'vra.domain.local'.
- SSO Default Tenant:** A text input field containing 'vsphere.local' and a 'Load' button to its right.
- Certificate:** A text input field and a 'Download' button to its right.
- View Certificate:** A blue hyperlink.
- Accept Certificate:** An unchecked checkbox.
- SSO Administrator Credentials:**
  - User name:** A text input field containing 'administrator@vsphere.local' and a 'Test' button to its right.
  - Password:** A text input field containing '\*\*\*\*\*'.
  - Confirm:** A text input field containing '\*\*\*\*\*'.
- IaaS:**
  - IaaS Server:** A text input field containing 'vRAiaaS.domain.local' and a 'Test' button to its right.
  - Below the field is the instruction: 'Enter the host name or IP address of the local machine.'

At the bottom right of the window, there are three buttons: '< Back', 'Next >', and 'Cancel'.

10. The final screen shows you the summary page. Hit the **Install** button, and relax. You deserve it. When you return, the IaaS components will be installed. Give the server a reboot. Once it is back up, reboot the vRealize Automation appliance as well.

If you decide to watch the progress on the screen, you may notice references to VCAC and DynamicOps. It is worth pointing out that you will see many more references to VCAC and DynamicOps as we progress through the chapters. Please understand these are not typos or mistakes. It is legacy code and the references are not worth VMware's time to clean up as it is likely to be a huge undertaking.

After five minutes, you can log back into the vRealize Automation Appliance via `https://vRA.domain.local:5480`. On the **Services** tab, you should see 22 services and they should all have a **Registered** status.



Note that when you reboot the vRA Appliance, it can take up to five minutes for all of the services to register. If you login as soon as the appliance comes back up, you could perceive that there is an issue with your vRealize Automation environment, when all it really takes is patience.

## Upgrading the vRealize Automation and Identity Appliances

VMware releases small versions of the core components of vRealize Automation 6.2, as bugs are patches and other features are rolled in. The upgrade process is fairly easy and should be explained before we set up the vRealize Automation web portal. Follow these steps to upgrade to the next minor version. As an example, if you wanted to upgrade from 6.2.0 to 6.2.1, perform the following steps:

1. Open a web browser on a machine that is on the same subnet as the vRealize Automation appliance. This ensures there are no firewalls in place. In our example, this would be `https://vRA.domain.local:5480`.
2. Select the **Update** tab.
3. Under the **Actions** heading on the right-hand side of the screen, select **Check for Updates**.
4. If an update is discovered, click on **Install Updates** to apply the settings. Your appliances must have Internet access to perform this step. This can take upwards of 30 minutes to complete, so be patient.
5. Login to `https://vRAso.domain.local:5480` and repeat the same steps for the Identity Appliance. Reboot both appliances after completion.



Often, the incremental updates are to patch for vulnerabilities, such as Shellshock and Heartbleed, which were big exploits and were discovered and patched in 2014. It is worth mentioning because when you get to the point of updating the IaaS components on the Windows Server 2012 R2, it might not be necessary, as these components may not be affected by the patched vulnerability.

6. Next, RDP into the `vRAiaas.domain.local` server, open a web browser, and head back to `https://vRA.domain.local:5480/installer`.

7. Download the `DBUpdate.zip` file and extract the contents to a folder on your desktop.
8. Open a command prompt as an administrator. Depending on your UAC settings, you may get a pop-up confirmation box. Hit **Yes** and continue.
9. Run the following command:

```
DBUpgrade -S sql01.domain.local\MSSQLSERVER -d vCAC -E
```

This is the location of the SQL database. If the command fails, remove the `MSSQLSERVER`, SQL instance name, from the command and try again.

10. Once you have successfully updated the database, repeat the process that you followed in the previous section to upgrade the IaaS components.



When running the IaaS executable, if you are not presented with a **Detected Components** screen, do not worry. This means that the upgraded components did not extend to the IaaS software in Windows and can be safely skipped.

## Configuring vRealize Automation

Now, we can login to the vRealize Automation web portal and continue the configuration process. Follow the example in this chapter, let's continue by opening up a web browser and heading to `https://vra.domain.local/vcac`. You will see a screen similar to the following:

vmware

User name: administrator@vsphere.local

Password: ●●●●●●●●

Use Windows session authentication

Login

VMware vCenter Single Sign-On

Don't be alarmed that if the screen displays **vCenter**. Once authenticated, you will be redirected to the **vRealize Automation** page.

You will login with the `administrator@vsphere.local` account and the SSO password you have been using so far.



Once we configure the **Tenant** in the upcoming steps, we will be able to place a tick in the box and use **Windows Session Authentication** to login. You will also have to install the VMware Client Integration Plugin.

You will default to the **Tenants** screen, as shown in the following illustration. We do not want to add a new **Tenant**, but modify the `vsphere.local`, one that already exists. Click on the hyperlink for `vsphere.local` and you will be taken to an **Edit Tenant** page.

Click on the **Identity Stores** tab, and then click on the plus sign button named **Add Identity Store**.

Fill out the required contents and be sure to hit the **Test Connection** button. Once the test connection is successful, click on the **Add** button:

The screenshot shows the 'Edit Tenant: vsphere.local' page in the vRealize Automation interface. A sidebar on the left contains navigation options: Tenants, Branding, Email Servers, Event Logs, and Advanced Services. The main content area displays the 'Add Identity Store' configuration form. The form includes the following fields:

- Name:** My Domain
- Type:** Active Directory
- URL:** ldap://domain.local:389
- Domain:** domain.local
- Domain alias:** DOMAIN
- Login user DN:** CN=vcs,CN=Users,DC=domain,DC=local
- Password:** (masked with dots)
- Group search base DN:** CN=Users,DC=domain,DC=local
- User search base DN:** CN=Users,DC=domain,DC=local

At the bottom right of the form, there are three buttons: 'Test Connection', 'Add', and 'Cancel'. The 'Test Connection' button is highlighted in blue.



There are a couple of items in the preceding screenshot that you need to pay special attention to. The **Domain alias** should be the short name you use to login to your Active Directory domain. Make sure you do not use this name in the **Name** field, as it will cause conflicts. In certain Active Directory environments, **Users** and **Groups** are kept in **User** container, which is a Microsoft default. Finally, it is recommended that you use a service account in the **Login user DN** field.

Click on the **Administrators** tab, and search for users in your Active Directory domain. Here, you will be able to define **Tenant Administrators** and **Infrastructure Administrators**.

Tenant administrators are responsible for creating business groups and designating a manager to each of these groups. Also, the Tenant administrator can create blueprints and publish catalog items.

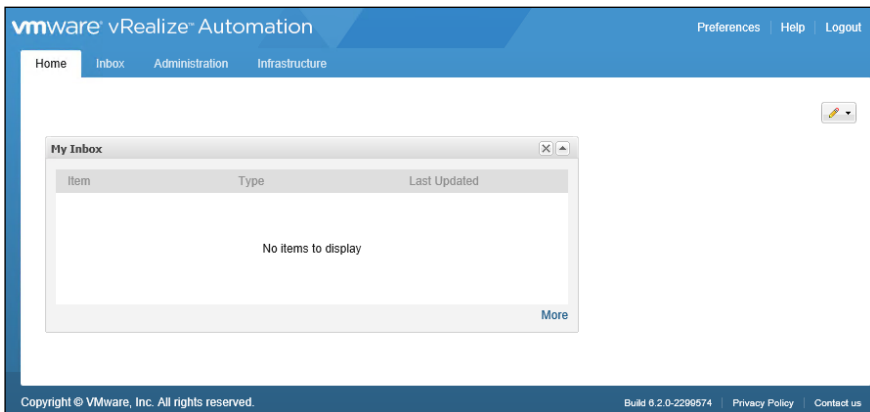
Infrastructure administrators have additional roles. They configure the endpoints in a web portal, organize compute resources, create and assign fabric groups, and create reservations and reservation policies.

You are likely to have noticed that there are other tabs and configuration options within the portal. However, we will configure this later as a member of the **Infrastructure Administrator** group.

Once these items have been defined to your satisfaction, you may logout of the web portal.

## Configuring the vRealize Automation infrastructure

Let's log back into the web portal, but this time as one of **Infrastructure Administrators** you assigned out of Active Directory in the previous section. For this example, we will login by placing the tick in **Use Windows Session Authentication**. Once logged in, you should see a screen similar to the following:



We will go over the customization options of this first screen and others in *Chapter 10, Customizing the End User Portal Experience*. For now, let's click on the **Infrastructure** tab and continue setting up the web portal.



Unless explicitly stated otherwise, for this section of the book we will be working exclusively under the **Infrastructure** tab.

The first thing we need to do is enter the license key for the second time. You can do this by performing the following steps:

1. Click on the **Infrastructure** tab, then on **Administration** on the left-hand side, and finally on **Licensing**.
2. In the upper right-hand corner, click on **Add License**.
3. You can cut and paste the license or type it in. Once done, hit **OK**.




The license key you enter dictates whether you will be able to use the features associated with Advanced or Enterprise editions of vRealize Automation. There is no need to download new software if you are going to upgrade versions.


4. Next, we need to set up some credentials so we can import vCenter information.
5. Under the **Infrastructure** tab, click on **Endpoints**, which is on the left-hand side, and then click on **Credentials**. Click on **New Credentials** in the upper right-hand corner and add the service account you use to manage your vCenter environment.
6. In the **Name** Field, provide the friendly name of your vCenter service account. We will use `vcs`. Username must be defined in this format: `vcs@domain.local`. Finally, add the user password, then click on the green check mark button to the left of the **Name** field.



7. For each endpoint you plan on utilizing through vRealize Automation, you will need to enter credentials. The exception is if you have multiple endpoints that work with the same account. In our example, this means that multiple endpoints can utilize the `vcs@domain.local` account.
8. Now that we have a user account defined that can manage vCenter, lets add vCenter as an endpoint.
9. Click on **Endpoints** on the left-hand side. In the upper right-hand corner, select **New Endpoint**. Drop down to the **Virtual** category at the bottom, and then select **vSphere (vCenter)**.
10. On the next screen, enter **Name** and **Address** of vCenter. Also, browse for the credentials you just added. The vCenter address should be in this format: `https://vcenter.domain.local/sdk`. Be sure to place a check in the box next to specify manager for network and security platform. You will need to enter the FQDN of your vCenter server and add the same credentials again. The FQDN would be `https://vcenter.domain.local` in our example. When this is done, hit **OK**.

 Failure to select the checkbox and enter the credentials will leave the **Compute Resources** portion blank, and you will be unable to use the vRealize Automation web portal.

11. Once our endpoint for vCenter is added, we need` to give vRA time to import all its settings. We can check vRA's progress by clicking on **Infrastructure** and the **Compute Resources** link on the far left-hand side.
12. Hover, your mouse over the vCenter environment listed in the center of the screen and select **Data Collection** at the bottom of the floating menu. This will list the status of the various areas of the individual endpoint as shown in the following figure:

 It is highly recommended that you make sure all the sections listed in this section have been completed successfully before continuing to configure your environment.

The screenshot shows the vRealize Automation console interface. At the top, there is a navigation bar with tabs: Home, Catalog, Items, Requests, Inbox, Advanced Services, Administration, and Infrastructure. Below the navigation bar, a breadcrumb trail reads: You are here: Infrastructure > Compute Resources > Compute Resources.

On the left side, there is a sidebar menu with the following items: < Back to Infrastructure, Compute Resources (highlighted), Cost Profiles, and EBS Volumes.

The main content area is divided into three sections:

- Inventory:** Last completed: 11/12/2014 1:36 PM, Status: Succeeded, Data collection:  On  Off, Frequency (hours): [dropdown menu] (Leave blank for daily data collection), Request now button.
- State:** Last completed: 11/12/2014 3:23 PM, Status: Succeeded, Data collection:  On  Off, Frequency (minutes): [dropdown menu] (Leave blank for data collection every 15 minutes), Request now button.
- Performance:** Last completed: 11/12/2014 1:45 PM, Status: Succeeded, Data collection:  On  Off, Frequency (hours): [dropdown menu] (Leave blank for daily data collection), Request now button.

So far in this section, we have added the vCenter service account and password, as well as imported our vCenter environment as an endpoint in vRealize Automation 6.2. Next, we want to configure groups. We will configure the **Fabric Group** first.

1. Click on **Groups** on the left-hand side, then configure **Fabric Group**.
2. Click on **New Fabric Group** in the upper right-hand corner. You will need to provide a name for the **Fabric Group**, as well as **Fabric Administrators**. It is suggested to name fabric groups in a fashion that correlates with the compute resources they will manage. Since we have been talking about vCenter this far in the book, you could name your **Fabric Group** as vCenter Fabric Group. If you do not see the vCenter endpoint we added under **Compute Resources**, you may need to wait a few minutes, or login to your vRealize Automation IaaS server and restart all of the vRealize Automation services.

3. Place a tick in the box next to your vCenter endpoint under the **Compute Resources** section, and then hit **OK**.



If you are stuck at this point, please flip ahead to *Chapter 11, Troubleshooting vRealize Automation 6.2* for some common fixes that can put you back on track.

Before we configure **Business Groups**, we need to create a **Machine Prefix**. While it may seem strange not to configure a **Business Group** at this point (after all, you are still in the **Groups** category if you are following along), **Business Groups** have a dependency on **Machine Prefixes**.

It is true that we could create the **Machine Prefix** from the **Business Groups** screen, while we are getting comfortable navigating around the web portal, it makes sense to create it in its own designated section.

Click on the **Back to Infrastructure** breadcrumb link, then on **Blueprints**, and **Machine Prefix**. As we have discussed previously, **Machine Prefixes** is the naming convention you or your enterprise uses to name machines provisioned in vCenter. It is recommended that you create as many different ones as you need. In the design exercise in the previous chapter, we used these naming conventions to differentiate between Windows and Red Hat virtual servers: AE-Win-Test and AE-RH-Test. The **Number of Digits** field dictates how many numbers will be added to the end of the name string. For example, providing two digits will allow vRealize Automation to provision up to 99 individual machines under this particular **Machine Prefix** naming scheme. The **Next Number** field is specified to instruct vRA which number to use next. Since we are using a small infrastructure setup in the example of this book, two digits and the number 1 will work for our naming scheme. Be sure to save your changes by hitting the green arrow button next to the **Machine Prefix** you've created.



It is worth mentioning there is a great article and script written on this subject to help you use your company's host naming standards in more detail. It can be found at <http://dailyhypervisor.com/vcloud-automation-center-vcac-5-2-custom-hostnaming-extension/>.

---

Now, we can finish configuring **Groups**. Click on **Infrastructure**, then **Groups** on the left hand-side, then configure **Business Groups**. Click on **New Business Group** in the upper right-hand corner. You will need to provide a name for the **Business Groups**, as well as select the default **Machine Prefix**. You will want to add the manager of this business group in the **Group Manager Role** field. You will also want to define an e-mail address to send manager e-mails. It could be an individual e-mail address or a distribution group. This is a required field. You can define individuals in the **Support Role** section, who will be points of contact for this business group. Depending on your company setup, this may be the IT Department. **Support Role** provides value so users in this group can emulate the end user experience. They can shadow the end user as well.

Finally, under the **User Role** section, add all of the users who will be in this **Business Group**. It is suggested to name **Business Groups** in a manner that it correlates with the compute resources they will manage. Since we discussed ACME Enterprises and the Software Development Group in the previous chapter, you could name your **Business Group** as `Software Development Group`. Once you have filled out all the information, hit **OK**.

You may have thought, "How will e-mails reach users? I don't recall setting up an integration with my e-mail server." Good catch. We have not set it up, but now is a good time to do so. Follow these steps to add an inbound and outbound e-mail connection to the vRealize Automation 6.2 web portal.

We are now leaving the **Infrastructure** tab portion of vRealize Automation web portal and will now be working in the **Administration** tab:

1. Click on the **Administration** Tab, then click on **Notifications** on the far left-hand side.
2. Click on **Email Servers**. You will have to add two. This does not refer to two e-mail servers, but only connections. One for inbound notifications and the other for outbound. You can only set up a total of two of these.
3. Click on the **Add** button. The contents of this section depend on several factors that are determined by the e-mail administrator. If you happen to be in this role, you should not have any issues filling out the information needed.

Here is a screenshot of the inbound and outbound settings as a reference:

**Edit Inbound Email**

<b>* Name:</b> Exchange Inbound	<b>Description:</b>
<b>Security:</b> <input type="checkbox"/> Use SSL	
<b>* Protocol:</b> <input type="radio"/> IMAP <input checked="" type="radio"/> POP3	
<b>* Server Name:</b> 192.168.1.2	<b>* User Name:</b> vrac.email@domain.local
<b>* Server Port:</b> 110	<b>* Password:</b>
<b>Folder Name:</b>	<b>* Email Address:</b> vrac.email@domain.local
<b>Processed Email:</b> <input checked="" type="checkbox"/> Delete From Server	<b>Accept Self Signed Certificates:</b> <input type="radio"/> Yes <input checked="" type="radio"/> No

Test Connection Update Cancel

**Edit Outbound Email**

<b>* Name:</b> Exchange Outbound	<b>Description:</b>
<b>* Server Name:</b> 192.168.1.2	<b>Authentication:</b> <input type="checkbox"/> Required
<b>* Encryption Method:</b> <input type="radio"/> Use SSL <input type="radio"/> Use TLS <input checked="" type="radio"/> None	<b>User Name:</b>
<b>* Server Port:</b> 25	<b>Password:</b>
	<b>* Sender Address:</b> vRAC.Alerts@domain.local
	<b>Accept Self Signed Certificates:</b> <input type="radio"/> Yes <input checked="" type="radio"/> No

Test Connection Update Cancel

Be sure to test the connections before applying your settings. Also, make sure you use a public routable (a real e-mail address) as **Sender Address** in the outbound configuration. When we cover **Approval Policies** later, you will see how to permit user requests to provision machines via e-mail. This means that if you are away from your desk or office and a provisioning request comes in, you can respond to the e-mail and the machine will provision it as the end user has requested.

Let's now go back to the **Infrastructure** tab, and build the last few remaining items.

We need to setup **Network Profiles** and **Reservation Policies** before setting up a **Reservation**.

To set up **Network Profile**, click on **Infrastructure | Reservations | Network Profile**. We will set up two private **Network Profiles**. One will be for DHCP, the other for a specific range of static IP addresses.

Hover your mouse over **New Network Profile** in the upper right-hand corner and select **Private** from the drop-down menu. Fill out the required fields using this screenshot as a reference. Hit **OK** when you've finished:

**New Network Profile - Private**  
Create a network profile to manage ranges of static IPv4 network addresses.

**Network Profile Information** **IP Ranges**

\* Name: DHCP

Description:

\* Subnet mask: 255.255.255.0

\* Gateway: 192.168.1.1

**DHCP**

Enabled

\* IP range start: 192.168.1.100

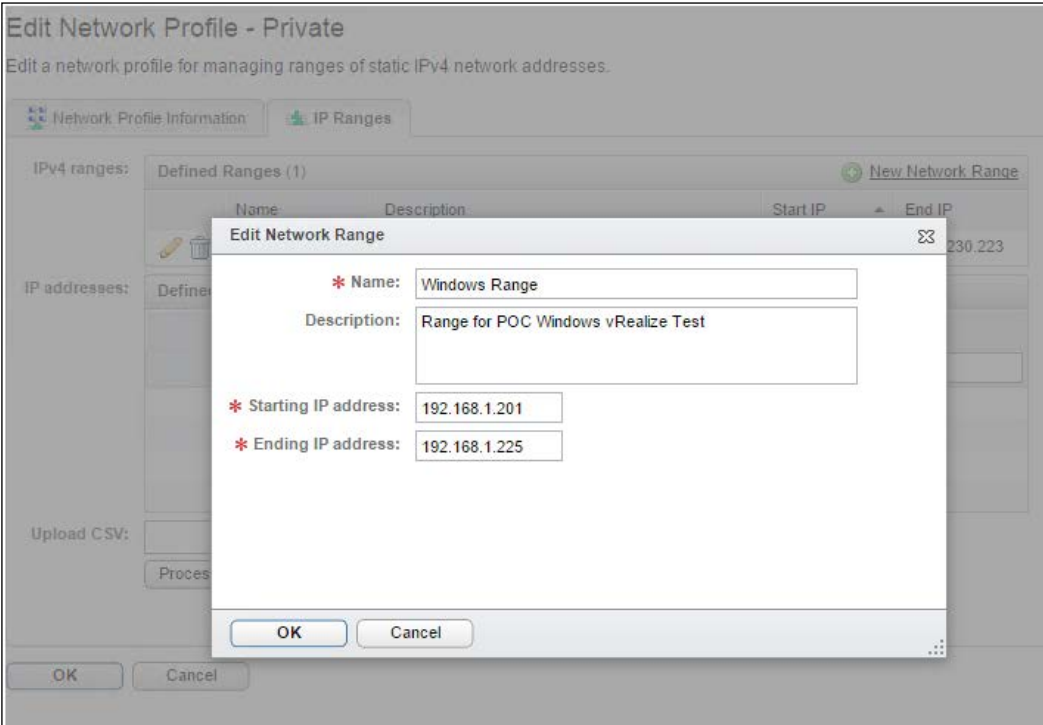
\* IP range end: 192.168.1.200 !

Lease time (seconds): Unlimited

OK Cancel

Repeat the preceding process, but this time we are going to define a limited number of static IP addresses to be assigned. As we discussed previously, it is quite unlikely to have a DHCP server in a production environment, serving up IPs to servers. We will have to define the IP addresses in this scenario. We will skip the section under the DHCP area, and click on the **IP Ranges** tab. Note, you will still have to provide **Name**, **Subnet mask**, and **Gateway** before proceeding to the **IP Range** tab.

Click on the **New Network Range** button. Provide the information for your environment and reference the following screenshot as an example:



Once you hit **OK**, the IP addresses section will be populated with the individual IP addresses defined in the preceding range. Hit **OK** to save your **Network Profile**.

Let's create **Reservation Policy**. We will create a reservation policy called **Software Developers**. Also, we will create a **Storage Reservation Policy**. We discussed in the previous chapter that it may make sense from a design perspective to create **Bronze**, **Silver**, and **Gold Storage** reservation policies based on available disk (such as SATA, SAS, and SSD). For our example, we will create a single **Storage Reservation Policy**.

---

First, click on **New Reservation Policy** in the right-hand corner and name it `Software Developers`. Click on the green check button to save.

Next, click on **New Storage Reservation Policy** and name it `vCenter Storage`. Click on the green check button to save.

Lastly, we need to create **Reservation**. Previously, we noted that this is a key design element and by this point, you should have an idea of what portion of your vCenter infrastructure you want to carve out for provisioning through vRA.

Click on the **Infrastructure** tab, and then click **Reservations** on the left-hand side of the web portal.

Hover your mouse over the **New Reservation** button in the upper right-hand corner. This will display a floating menu. Select **Virtual**, then **vSphere (vCenter)**.

Under **Compute Resource**, you will use the down arrow to select the vCenter endpoint we set up earlier in the chapter. You will need to give the **Reservation** a name. Although this value will be prepopulated for you, we will name it `Software Developer Reservation`. **Tenant** will be filled out for you automatically and will contain `vsphere.local`. You are required to select a business group. Since we are dealing with a **Reservation** for software developers, we will choose the **Software Developers Business Group**.

**Priority** is important. It dictates which **Reservation** gets the highest priority in terms of provisioning compute resources for end user requests. This is helpful in the event multiple reservations have the same compute resources. The lower the number, the higher the priority. Since this is the first one we are creating, we will assign a **Priority** of 0.

Once you have the first section filled out with all the required options, click on the **Resources** tab to continue.

Under the **Resources** tab, you will see all of the memory and storage dedicated to your vCenter environment. It is here that you will need to define how much and what resources will be available for consumption for your software developer users. As you can see from this screenshot, we allow a total of 32 Gig of RAM out of a total of 208 Gig available. This may seem like an odd number for the memory, but vRA has already excluded memory from the list that is already dedicated to your ESXi hypervisor:



**Edit Reservation - vSphere (vCenter)**  
 Modify the compute resource by making the following changes:

\* Memory: **Memory (GB)**

Physical	Reserved	Allocated	This Reservation
208	32	8	32

\* Storage: **Storage Paths (10)**

	Storage Path	Storage Cost (per GB)	Physical	Free	Reserved	This reservation reserved	This reservation allocated
<input checked="" type="checkbox"/>	7410_LUN1	\$0.0000	1024	297	40	40	
<input type="checkbox"/>	7410_LUN2	\$0.0000	1024	155	0		
<input checked="" type="checkbox"/>	7410_LUN3	\$0.0000	1024	190	150	150	
<input type="checkbox"/>	7410_Oralab	\$0.0000	1024	565	0		
<input type="checkbox"/>	7410	\$0.0000	1024	1023	0		
<input checked="" type="checkbox"/>	7410_vCloud	\$0.0000	1024	1022	750	750	
<input type="checkbox"/>	7410-LOG	\$0.0000	50	49	0		
<input type="checkbox"/>	Local_ESXL01	\$0.0000	62	61	0		
<input type="checkbox"/>	Local_ESXL02	\$0.0000	62	61	0		
<input type="checkbox"/>	TEMPLATE_ESX	\$0.0000	295	260	0		

Resource pool:

It is important to make sure you give enough resources for consumption. If you are too conservative, you may run into errors related to a lack of compute resources when you deploy **Blueprints**. We will cover this in the next chapter. Click on **OK** and we will go over to the **Network** tab.

Under the **Network Section**, a list of all your vSwitches and vDistributed switches will appear. You can select as many as you need, but for this example, we will only select one. Also note, if you do not select **Network Profile**, vRA will use DHCP by default. If you do not have a DHCP server, and do not select **Network Profile**, reflecting a range of static IPs, the machine may fail to provision. When this is finished, hit **OK**.

It bears repeating that the **Alerts** tab is depreciated and will be going away in a future build of vRealize Automation. For this purpose, we will not spend any time covering its features, but feel free to explore its contents.

---

Let's take an opportunity to perform a sanity check before going into the next chapter:

- Double check and make sure time zones and the time is in sync between the vRA and Identity Appliance, as well as the IaaS server
- Make sure all the services on the vRA Appliance are registered
- Make sure all of the services related to vRA in the services snap-in on your IaaS server are running
- Make sure the process under **Compute Resources** have finished running
- You can test the status of vRA by viewing the XML contents of the server at `https://vRA.domain.local/component-registry/services/status/current`

These simple checks can help ensure you do not run into unexpected issues in the future.

## Summary

This is where we got to apply the lessons we learned in the previous chapters and install the vRealize Automation 6.2 core components. We followed step-by-step instructions to set up the core components and highlighted some of the potential stumbling blocks that could cause frustration.

After the core components were installed, we focused on configuring the basic infrastructure of vRA, so we could provide prepare machine provisioning for end users.

In *Chapter 5, Mastering Blueprints*, we will continue our journey in mastering vRealize Automation 6.2 by taking a deep dive into creating Blueprints.



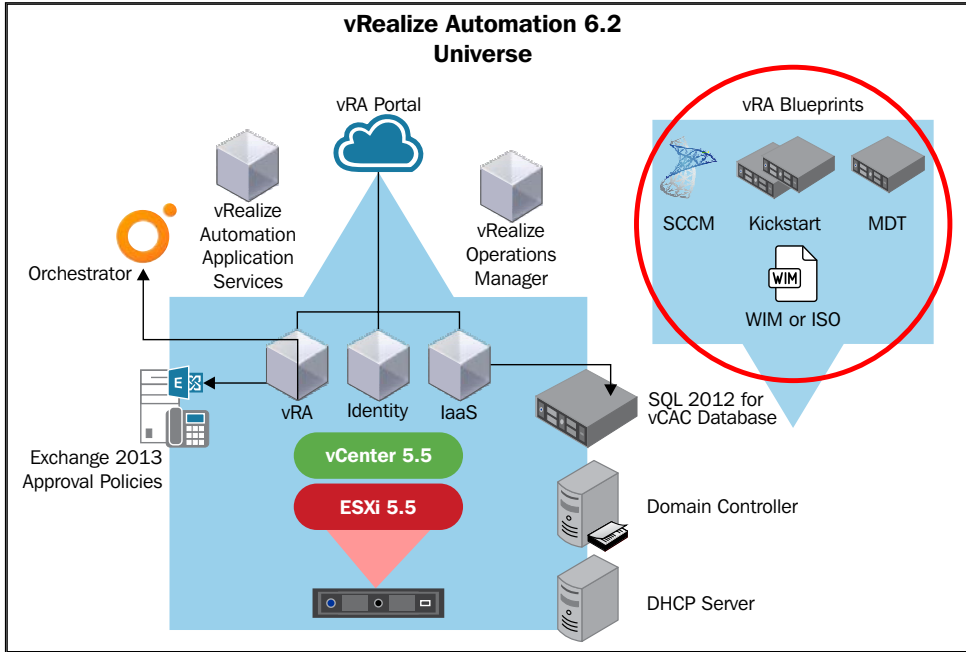
# 5

## Mastering Blueprints

There is little doubt we have covered a wealth of information in the previous chapter. As stressed before, there are a lot of components in vRealize Automation and a lot of ground to cover before you can start to unlock its potential. By now you should have a basic understanding of Blueprints and their role in the vRA universe. In this chapter, we are going to deep dive into Blueprints. We will set up Blueprints to deploy from vCenter snapshots, templates, WIM, SCCM, and Kickstart. We will discuss the following topics in this chapter:

- Creating a linked clone Blueprint
- Creating a clone Blueprint
- vRA guest agents
- Creating a WIM Blueprint
- Creating a Blueprint with MDT
- Creating a SCCM Blueprint
- Creating a Kickstart Blueprint
- Multi-machine Blueprints
- Provisioning physical machines
- Property dictionary
- Destroying a provisioned machine

We will now focus on the following components of the vRA Universe:



It is worth noting that you cannot edit any of the workflows that will be discussed in this chapter. You can create your own workflows in vRO or vRealize Automation Designer. You can also inject your workflows into the out of the box ones provided with vRealize Automation. We will cover vRO in *Chapter 8, REST API and vRealize Orchestrator*.

## Creating a linked clone Blueprint

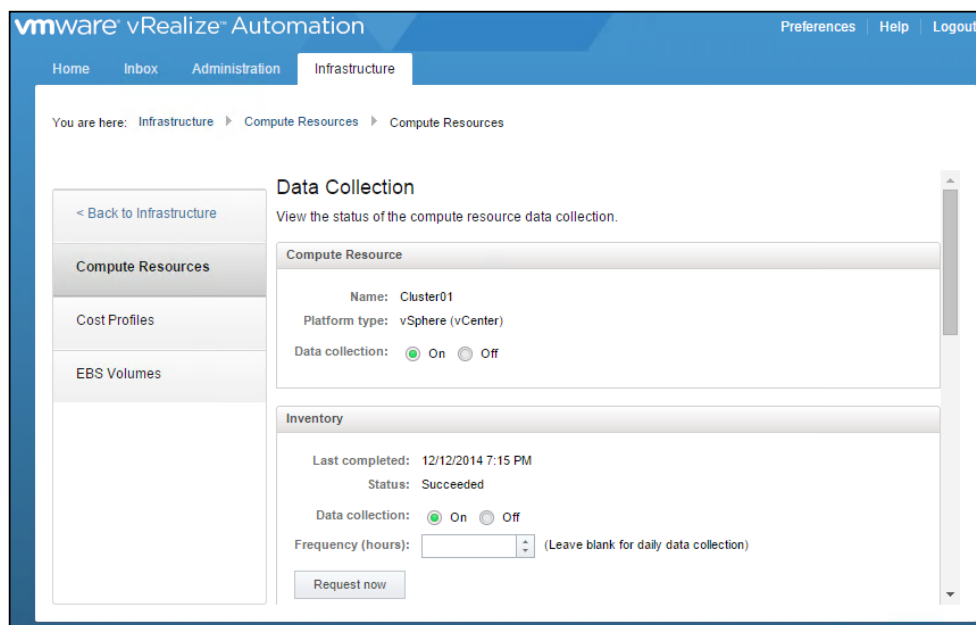
To start us off on our journey of providing infrastructure to end users, let's start with the most basic example: **creating a linked clone Blueprint**. A linked clone is a copy of a VM parent and shares its virtual disks, which allows it to conserve space. Since we save space through the usage of linked clones, we can provision more VMs.

For this example, we will create a Blueprint and deploy Windows Server 2012 R2. However, before we continue, make sure you have a snapshot of a VM and a template created in your vCenter environment. We will be referring to them in the next two examples.

By default, the vRealize Automation inventory service only runs once every day. In order to see newly created snapshots and templates, we must run the inventory service manually.

To do so, login to the vRealize Automation web portal:

1. Go to the **Inventory** tab, then click on **Compute Resources**. Click on **Compute Resources** once again and you will see the resources in the main workspace. Hover your mouse over the **Compute Resources** and a drop-down menu will appear.
2. Select **Data Collection**. Under the **Inventory** section, hit the **Request Now** button. It could take 5-10 minutes to complete, depending on the size of your environment. You can also change the default inventory frequency by modifying the **Frequency** field. Take a look at this example:



3. Once this task is complete, click on the **Infrastructure** tab, then on **Blueprints**.

4. Click on **Blueprints** once more, which is on the left-hand side, and the center of your screen should be blank, as we have not created any Blueprints yet.
5. Hover your mouse over the **New Blueprint** button, then select **Virtual** and **vSphere (vCenter)**.

Your screen will be similar to the following:

**New Blueprint - vSphere (vCenter)**  
Create a blueprint to define a catalog item for infrastructure provisioning. You also can copy an existing blueprint to use as a starting point.

Copy from existing blueprint: -- Select an item to copy --

**Blueprint Information** | Build Information | Properties | Actions

\* Name:

Description:

Blueprint options:

- Master (can be copied)
- Display location on request
- Shared blueprint (can be shared across groups)

Reservation policy:

\* Machine prefix: Use group default

Maximum per user: Unlimited

\* Archive (days):

Cost (daily):

OK Cancel

Under **Blueprint Information**, in the **Name** field, provide a name, such as `Windows 2012 R2 Linked Clone`. Under **Blueprint** options, we have the following:

- **Master:** This means that we can use this as a base Blueprint for future instances
- **Display Location on Request:** This option needs to be checked if you are going to deploy the Blueprint in multiple locations
- **Shared Blueprint:** This is selected by default and it means that the Blueprint can be shared across multiple business groups

You can select **Reservation Policy**, as was defined and created in the previous chapter. However, this is not necessary. Under **Machine Prefix**, let's select the one we created for Windows previously, `AE-Win-Test-`.

**Maximum per user** is the field where you can define how many machines a user can provision from this Blueprint. This is a key element that you will want to consider as you build your vRA environment.

**Archive (days)** refers to how long you will keep the virtual machine after expiration. Setting this to 0 means it will not be archived.

Finally, you can set a cost per day (though it's not required) for this particular Blueprint. You can set costs on a compute level under **Infrastructure | Compute Resources | Cost Profiles**.

Click on the **Build Information** tab and select the drop-down arrow beside **Action**. This will show you a list of the built-in workflows provided. Take a look at the following screenshot:

**New Blueprint - vSphere (vCenter)**  
Create a blueprint to define a catalog item for infrastructure provisioning. You also can copy an exist

Copy from existing blueprint: -- Select an item to copy --

Blueprint Information | **Build Information** | Properties | Actions

Blueprint type: Server

Action: **Create**

\* Provisioning workflow: Create  
Clone  
Linked Clone  
NetApp FlexClone

Machine Resources

	* Minimum	Maximum ⓘ
CPUs:	<input type="text"/>	<input type="text"/>
Memory (MB):	<input type="text"/>	<input type="text"/>
Storage (GB):	<input type="text"/>	<input type="text"/>
Lease (days):	<input type="text"/>	<input type="text"/>

(Leave blank for no expiration date.)

For this example, we will select **Linked Clone**. Doing this will automatically fill in **Provisioning workflow** with **Clone Workflow**.

Next, you will need to hit the button to the right of **Clone From**. This will allow you to browse your vCenter environment and choose a VM.



Select a VM from your vCenter environment that already has a snapshot created. If there isn't one, go into vCenter and take a snapshot. You can take one from within vRA, but it has to be from a machine already managed by vRA. This is controlled by **Infrastructure Manager**, which we discussed in *Chapter 2, vRealize Automation Architecture and Feature Overview*.

Once your snapshot has been selected, you will have to assign machine resources. It is a best practice to select the bare minimum. You can always increase the resources if needed, but you cannot take them away. If you create a Blueprint with too many resources, there will be no way to change this after the fact. You will have to create a brand new Blueprint. This gets even more complicated if you have deployed VMs from this particular Blueprint. You cannot destroy a Blueprint if you have VMs that have been provisioned from it without first destroying the VMs.

Go to the **Properties** tab and under **Custom Properties**, simply add the following: **VMware.VirtualCenter.OperatingSystem** and give it Value of **windows8Server64Guest**.

Remember that these components are case sensitive. At a bare minimum, you have to provide the **VMware.VirtualCenter.OperatingSystem** property. Once complete, hit **OK**. Congratulations, you have just created your first Blueprint.

There are a lot of custom properties created by VMware for use in vRealize Automation. We will go over many of them in this chapter. Please see *Chapter 12, References for vRealize Automation 6.2* for a list of all the custom properties.

Now, let's publish it. Without performing this step, your Blueprints will not be available for users to leverage and provision VMs. Hover your mouse over **Blueprint** and select **Publish** from the floating menu.

You will be redirected to a confirmation screen. Hit **OK** and your Blueprint will be published.

Let's set up the remaining parameters necessary to test our work. Once complete, we will come back and build other Blueprints and discuss the customizations.

## Preparing the linked clone Blueprint for provisioning

We will need to build more of the vRA web portal so that we can deploy the Blueprint we've just created:

1. Click on the **Administration** tab, then **Catalog Management**. We will need to create a new service.


2. Click on **Services**, then click on the **Add** button.
3. Let's give the service a name, such as `Windows`. You can select an icon to represent the service, as well as define properties, such as the **Owner**, **Support Team**, and **Change Window**. Take a look at the following screenshot:

**Add Service**

\* Name:

Description:

Icon:

Preview: 

Status:

Hours:  GMT-05:00 to  GMT-05:00

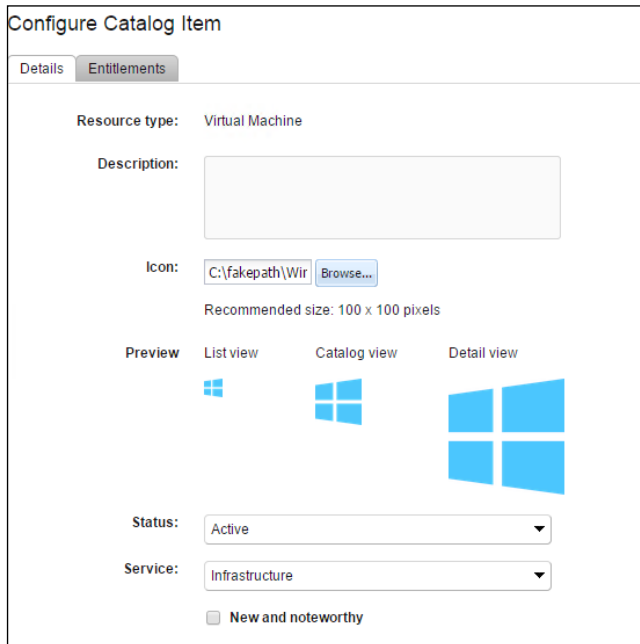
Owner:

Support Team:

Change Window:  GMT-05:00 to  GMT-05:00

4. Make sure to make the **Status Active** by selecting this option from the drop-down menu. Once complete, hit the **Add** button.
5. Next, select **Entitlements** on the left-hand side of the web portal. Click on the **Add** button.
6. Provide a name for your entitlement. Let's continue using the software developers as our example. Type them in the **Name** field.
7. Select the software developers from the Business Group drop-down menu.
8. Click on the **Items and Approvals** tab. This is where you can link what a user does with a catalog item. Under **Entitled Services**, click on the green plus sign. This will display the **Service** we just created titled `Windows`. Select this option and you will be returned to the previous screen.
9. Repeat this process for **Entitled Catalog Items**. Select the Blueprint we created previously.

10. Lastly, repeat this process to add **Entitled Actions**. You will likely want to spend some time going over all the actions that can be assigned. Since this is an example for learning purposes, let's select all the available options. Click on **Update** once complete.
11. Next, select **Catalog Items** on the left-hand side of the web portal. You should see the Blueprint item you just published. Click on the item and you will see a screen similar to this below:



12. Make sure to make the **Status Active** and under **Service**, select the **Windows** option (at this point, it should be the only one you've created).
13. Click on the **Entitlements** tab and you should see the entitlement you created in the previous step. Click on **Update**.

## Provision the linked clone Blueprint

We have created a linked clone Blueprint for Windows Server 2012 R2. We have also made the necessary settings under the **Administration** tab so that we can deploy the Blueprint.

Since we have not configured any approval policies (we will cover this in the next chapter), we will be able to have a member of the software developers team login to the vRA web portal and submit a request to deploy the Blueprint.

Once the software developer logs in, he or she would click on the **Catalog** tab. On that page the Windows Blueprint we created will be displayed. Hit the **Request** button and you will be redirected to a page that displays the options available. To illustrate this, take a look at the following screenshot:

The screenshot shows a web interface for requesting a resource. At the top, there is a table with columns: Blueprint, Description, Machines, and Daily Cost. The first row shows 'Windows 2012 Linked Clone', an empty description, '1' machine, and '\$0.00' daily cost.

Below the table are two tabs: 'Request Information' (selected) and 'Storage'. Under 'Request Information', there are several input fields:

- Machines:** A dropdown menu set to '1'.
- CPUs:** A dropdown menu set to '1' with a note '(Select 1-4)'.
- Memory (MB):** A dropdown menu set to '2048' with a note '(Select 2048-4096)'.
- Storage (GB):** A dropdown menu set to '100'.
- Description:** An empty text area.
- \* Owner:** A text field containing 'software.developer@domain.local' with a search icon to its right.
- Reason for request:** An empty text area.

At the bottom right of the form, there are three buttons: 'Save', 'Submit', and 'Cancel'.

If you want to increase **CPU**, **Memory**, or **Storage**, you could do so at this stage. When we discuss **Approval Properties** in the next chapter, we will explain how to place checks and balances for these requests.

When you are satisfied with the parameters, hit the **Submit** button.

You will be redirected to a confirmation page. We will discuss how to track these requests from the portal in a later chapter. If you want a quick glance to see if this is working as expected, open the vSphere client and login to vCenter. After a few minutes of submitting your request, you should see the VM being provisioned.



If your linked clone Blueprint did not provision, please skip over to *Chapter 11, Troubleshooting vRealize Automation 6.2.*

# Creating a clone Blueprint

The process of creating a clone Blueprint is very similar to building a linked clone Blueprint. In fact, to create the next Blueprint, we will leverage an improvement introduced in the 6.0 series of vCloud Automation Center. It's called Copy from Existing Blueprint. We mentioned the **Master** option earlier in this chapter when we created the linked clone Blueprint. You need to have this enabled to use the feature. Make sure you have this option selected on the pervious Blueprint before continuing.

Click on the **Infrastructure** tab, then on **Blueprints**. Click on **Blueprints** once more on the left-hand side.

Hover your mouse over the **New Blueprint** button, then select **Virtual** and **vSphere (vCenter)**.

Your screen will look similar to the following:

**New Blueprint - vSphere (vCenter)**  
Create a blueprint to define a catalog item for infrastructure provisioning. You also can copy an existing blueprint to use as a starting point.

Copy from existing blueprint: -- Select an item to copy --

**Blueprint Information** | Build Information | Properties | Actions

\* Name:

Description:

Blueprint options:

- Master (can be copied)
- Display location on request
- Shared blueprint (can be shared across groups)

Reservation policy:

\* Machine prefix:

Maximum per user:

\* Archive (days):

Cost (daily):

OK Cancel

At the very top of this dialog box, you will see **Copy from existing blueprint**. Select this option and choose **Windows 2012 R2 Linked Clone**.

You will see by clicking through the tabs that all the options we previously defined are present in this Blueprint. This allows us to work in a faster and smarter manner.

Fill in the following fields like you did earlier:

- **Name**
- **Machine prefix**
- **Archive**

Click on the **Build Information** tab and select the drop-down arrow beside **Action**. This will show you a list of the built-in workflows provided, as displayed below:

Modify the blueprint by making the following changes:

Blueprint type: Server

Action: Clone

\* Provisioning workflow: Create

\* Clone from: Linked Clone

Customization spec: NetApp FlexClone

Machine Resources

	* Minimum	Maximum
CPUs:	1	4
Memory (MB):	2048	4096
Storage (GB):	100	125
Lease (days):		

(Leave blank for no expiration date.)

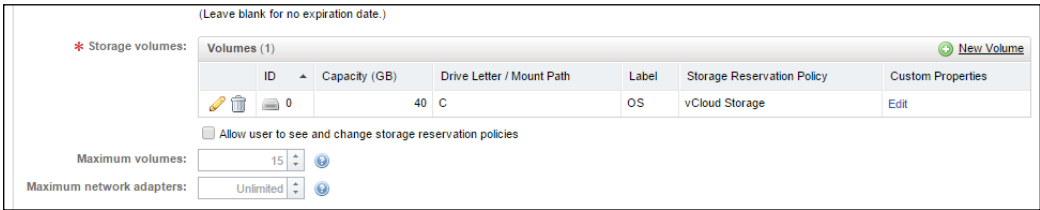
For this example, we will select **Clone**. Doing so will automatically fill in **Provisioning workflow** with **CloneWorkflow**.

The difference between a linked clone and a clone is that the former lets you provision from a snapshot, whereas the latter is a provision from a vCenter template.

Next, you will need to hit the button to the right of **Clone From**. This will allow you to browse your vCenter environment and choose an existing VM template.

You may have also noticed that there is a section at the bottom of this screen for **Storage Volumes**. This is a required field and must be completed. It is quite likely that it's already populated and all you have to do is specify the **Mount Path** by clicking on the pencil icon to edit it. If you need to create one, then under the **Storage Volumes** section, click on **New Volume**. This will allow us to add a C drive for Windows to be installed. You can also select **Storage Reservation Policy**.

Take a look at the screenshot below as an example:




Note that the drive letter is listed as C not C:

You can click on the **Properties** tab and see that the **Custom Property** we defined previously is still listed as **VMware.VirtualCenter.OperatingSystem** with a **Value** of **windows8Server64Guest**.

Remember that these components are case sensitive. At the bare minimum, you have to provide the **VMware.VirtualCenter.OperatingSystem** property. Once you complete this, hit **OK**. Don't forget to publish it.

Without performing this step, your Blueprints will not be available for users to leverage and provision VMs. Hover your mouse over **Blueprint** and select **Publish** from the floating menu.

You will be redirected to a confirmation screen. Hit **OK** and your Blueprint will be published.

 For each Blueprint created, you will need to configure or add it to an existing service and assign an entitlement. Since we have already covered how to do this, we will not repeat this step while creating each Blueprint.

## vRealize Automation guest agents

We have mentioned vRealize Automation guest agents previously, and now they finally come into play. While they're not needed for clone and linked clone Blueprints, they are needed for Kickstart, SCCM, and WIM deployments.

Guest agents allow customizations to be performed after the operating system has been deployed. Most importantly, the agents can signal back to the vRA web portal that the VM provisioning has completed.

---

In Windows, the guest is installed as a service and runs only once. This occurs once the base operating system has been applied. It allows tasks, such as sysprep and system rearm to occur. However, in the following examples, we will have these tasks completed by other software products, such as SCCM and MDT. Even so, we still need to slipstream the agent in, as it notifies vRA when the provisioning process is complete. Also, note that VMware tools is required as part of this process.

## Creating a WIM Blueprint

WIM Blueprints are great because they can tie into systems you may already have available, such as a **Windows Deployment Services (WDS)** or **Microsoft Deployment Toolkit (MDT)** server. However, from this point on, the Blueprints have many more components and, therefore, more potential for failure.

At a high level, WIM Blueprints work in this fashion:

- VMware WinPE ISO is mounted and boots a small Windows 2008 R2 based Kernel.
- This launches a command window that runs several VMware-specific batch scripts. These scripts request a DHCP address and try to connect back to your WIM location on the network via an encrypted network session.
- The WinPE ISO executes `diskpart` to create the partition for the OS to be installed.
- Finally, the WIM file is laid down on the partition as the Operating System.

Before we create the Blueprint, we need to generate the VMware WinPE Boot ISO

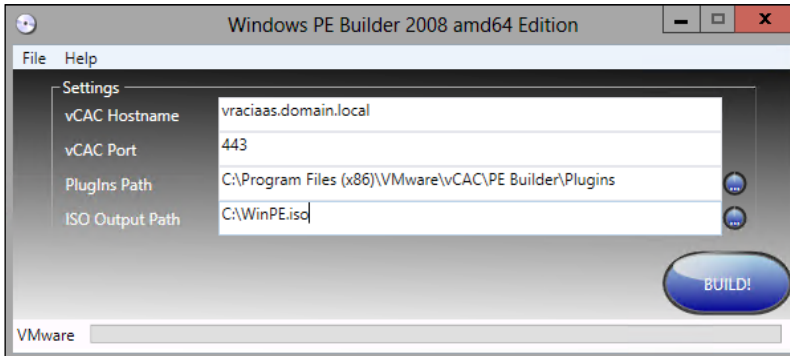
To do this, we will need to download and install the PE Builder Utility from <https://vRA.domain.local:5480/installer>. It is located under the **Provisioning Utilities** header.

This is Windows software and as such, it needs to be installed on a Windows server. Additionally, you will need to install the **Windows Automation Kit (WAK)** for Windows 8.1 or a later version of it. It is available at <http://www.microsoft.com/en-US/download/details.aspx?id=39982>.

Once you have installed the software, locate it on your computer. It should be called `PE Builder 2008`. It is a bit of a misconception to understand what information has to be added to the **vCAC Hostname** field. Our vRA host name in our examples has been `vRA.domain.local`. However, what is really needed in this field is the vRealize Automation IaaS FQDN server.



Take a look at the following example:



This will take quite a while to generate. When it's complete, upload the ISO to a shared storage device in your vCenter environment. For our example, we will be saving the ISO on a LUN titled `shared_storage`, in a folder called `ISO`. Please keep in mind that all the activities you perform in vRA are case sensitive. Take a look at the minimum custom properties you will have to define to get this to work. It is recommended to create a **Build Profile** out of these properties so you can use them with multiple Blueprints. Here is an example of the recommended properties:

Properties (13) <span style="float: right;">+ New Property</span>				
	Name	Value	Encrypted	Prompt User
	Image.ISO.Location	shared_storage	No	No
	Image.ISO.Name	/ISO/vRA_WinPE.iso	No	No
	Image.Network.Letter	Z	No	No
	Image.Network.Password	*****	Yes	No
	Image.Network.User	vcs@domain.local	No	No
	Image.WIM.Index	1	No	No
	Image.WIM.Name	Win2012R2.wim	No	No
	Image.WIM.Path	\\192.168.1.3\es\DeploymentShare\OperatingSystems\Win2012R2	No	No
	VirtualMachine.FloppyDrive.Attach	False	No	No
	VMware.Floppy.Drive.Mount	False	No	No
	VMware.Network.Type	VMXNET3	No	No
	VMware.SCSI.Type	IsiLogicSas	No	No
	VMware.VirtualCenter.OperatingSystem	windows8Server64Guest	No	No

Under the **Build Information** tab of your WIM Blueprint, you will select the **Action Create**. For **Provisioning Workflow**, select **WIMImageWorkflow** from the drop-down list.

Be certain to save your Blueprint; publish and test it before providing it to your end users.

---

# Creating a Blueprint with MDT

Using vRA with MDT is a great way to leverage automation that you may already be using in your environment. The same goes for SCCM and Kickstart. Why reinvent the wheel with vRA when we can leverage existing technologies? While we will not go into how to install and configure MDT (nor will we provide this information for SCCM or Kickstart), we will instruct you on how to tie it in to vRealize Automation.

Below, we will outline one of the more common ways to pull this off. First, let's cover at a high level what we are going to accomplish:

Notice this is very similar to a WIM deployment. At a high level, a Blueprint with MDT will work in this fashion:

- MDT ISO is mounted and boots a small Windows based Kernel.
- This launches a command window that runs several Windows MDT-specific scripts. These scripts request a DHCP address and try to connect back to your MDT server and performs tasks that you have defined, such as installing a Windows OS and third-party software.
- The task could also run scripts to harden your server; join it to the domain and activate it before completing this task.

Before we create the Blueprint, we need to update the MDT boot image and create an ISO. The easiest way to accomplish this task is to perform the following steps:































1. Open the MDT Workbench and click on the `Out-of-Box` folder.
2. Launch the **Import Driver Wizard** and import the INF files associated with VMware tools. The easiest way to accomplish this task is to install the VMware tools on a Windows-based VM, and point the Wizard to this location `C:\Program Files\Common Files\VMware\Drivers`.
3. You will want to browse through the subdirectories for the INF files in each category, such as `VMXNET3`. After all, it is a best practice to use the `VMXNET3` driver for all VMs in vCenter.
4. You will need to create a task that installs the vRA Guest Agent so that it is properly installed on the Windows OS. To do this, follow these steps:
  1. Download the guest software (64-bit for our examples) from <https://vRA.domain.local:5480/installer> and save the ZIP file on the MDT server in a folder under **Applications** called `VMware vCAC Agent`.
  2. Download the 32-bit command line version of 7zip. You will need this to get the Windows vRealize Automation Guest Agent installed successfully. It is available at <http://7-zip.org/download.html>.

3. Create a folder on the MDT server containing the GugentZip\_x64.zip and 7zip executable (7za.exe).
4. Create a batch file to do these steps and place it in the same directory as the other files in the previous step. Please note there is no space between -oC:

```
cmd /c mkdir C:\Windows\Temp
cmd /c copy
  "\\192.168.1.2\es$\DeploymentShare\Applications\VMware
  vCAC Agent\7za.exe" C:\Windows\Temp
cmd /c copy
  "\\192.168.1.2\es$\DeploymentShare\Applications\VMware
  vCAC Agent\GugentZip_x64.zip" C:\Windows\Temp
cmd /c "c:\Windows\Temp\7za.exe x -oC:\ -y
  GugentZip_x64.zip"
c:
cd C:\VRMGuestAgent
cmd /c "WinService.exe -i -h vRAiaas.domain.local:443 -p
  ssl"
cmd /c del /Q /S C:\Windows\Temp\7za.exe
cmd /c del /Q /S C:\Windows\Temp\GugentZip_x64.zip
cmd /c sc start VCACGuestAgentService
```

5. Create an application in MDT to install and run the batch script you just created.
  6. Run the unblock-file PowerShell command against all the files in the vRealize Automation Agent directory on the MDT server. You need to do this to make sure they are downloaded and installed on the provisioned OS correctly.
5. Update the **Deployment Share** in MDT, and make sure you generate a LiteTouch ISO file.

This will take quite a while to generate. When you've completed doing this, upload the ISO to a shared storage device in your vCenter environment. For our example, we will be saving the ISO on a LUN titled `shared_storage` in a folder called `ISO`. Take a look at the minimum custom properties that you will have to define to get this to work:

Properties (15) <span style="float: right;">New Property</span>				
	Name	Value	Encrypted	Prompt User
 	Image.ISO.Location	shared_storage	No	No
 	Image.ISO.Name	/ISO/LiteTouchPE_x64.iso	No	No
 	Image.Network.Letter	Z	No	No
 	Image.Network.Password	*****	Yes	No
 	Image.Network.User	vcs@domain.local	No	No
 	Image.WIM.Index	1	No	No
 	Image.WIM.Name	Win2012R2.wim	No	No
 	Image.WIM.Path	\\192.168.1.3\DeploymentShare\OperatingSystems\Win2012R2	No	No
 	VirtualMachine.FloppyDrive.Attach	False	No	No
 	VirtualMachine.Network0.MacAddress	00:50:56:87:	No	No
 	VirtualMachine.Network0.MacAddressType	static	No	No
 	VMware.Floppy.Drive.Mount	False	No	No
 	VMware.Network.Type	VMXNET3	No	No
 	VMware.SCSI.Type	lsiLogicSas	No	No
 	VMware.VirtualCenter.OperatingSystem	windows8Server64Guest	No	No

There are a few custom properties in this example that need an additional explanation so you understand how they tie in with MDT:

- We are using MDT's ISO to boot instead of VMware's WinPE ISO. This is reflected as `LiteTouchPE_x64.iso` in the preceding screenshot.
- To automate MDT tasks, we have to assign a serial or model number or MAC address in MDT. MDT looks for a unique setting to trigger the automated tasks. The model is not unique to VMs, and the serial number exceeds MDT's 15 character limit. That leaves us with the MAC address. Using the custom property, we can specify the MAC address to be used before the VM is created.
- **VMware.SCSI.Type** must be defined as `lsiLogicSas`.

Clone a copy of your WIM Blueprint. Under the **Build Information** tab of your newly created Blueprint, you will select the **Action Create**. For **Provisioning Workflow**, select **BasicWorkflow** from the drop-down list.

We will have to inject a vRO workflow to prevent a notification e-mail from going out as soon as the VM is powered on. This is one of the downsides of the Basic Workflow. However, once we add the vRO workflow, it will prevent the notification from going out until the MDT image is deployed successfully. We will add the file check workflow we create in *Chapter 8, REST API and vRealize Orchestrator* to this Blueprint.

Be certain to save your Blueprint, publish, and test it before providing it to your end users.

While MDT does not give you all the options of SCCM, it does make a great free tool to provision customized WIM files, while adding the additional ability to install software, Windows updates, and apply customized scripts.

## Creating a SCCM Blueprint

SCCM (**S**ystem **C**enter **C**onfiguration **M**anager) is an Enterprise level software program that comes as part of a greater System Center suite. For this book, we will be focusing on the 2012 R2 edition.

SCCM is a tool that not only deploys unattended, customized Windows images to virtual and physical endpoints, but it allows software and hardware inventory collection, license compliance, Windows software updates, and third-party software and scripts targeted to specific user or device containers. It is a versatile and amazing product.

It should be noted that the process to create a SCCM Blueprint is nearly identical to creating a MDT WIM Blueprint. At a high level, an SCCM Blueprint will work in this fashion:

- vRA creates a computer object in SCCM in the collection specified in the Blueprint or build properties.
- A computer object is created in Active Directory and added to the SCCM collection that we specify in the custom properties of the Blueprint.
- An SCCM ISO is mounted and boots a small Windows-based kernel.
- This launches a command window that runs several SCCM-specific scripts. These scripts request a DHCP address and try to connect back to your SCCM server and perform tasks you have defined, such as installing a Windows OS, and third-party software.
- The tasks could also run scripts to harden your server, join it to the domain, and activate it before completing.

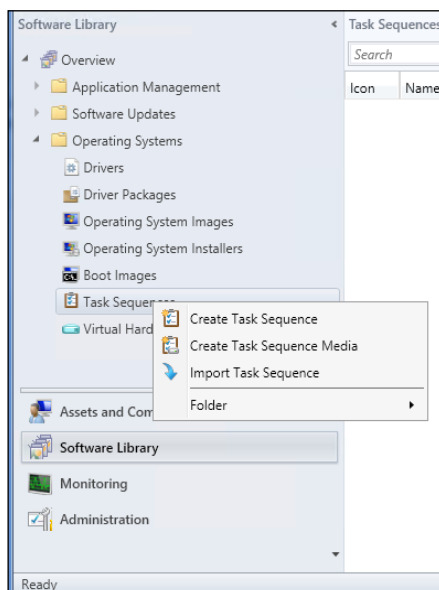
Before we create the Blueprint, we need to create an SCCM ISO. Although it is not in the scope of this book to provide instructions to deploy operating systems through SCCM, you must perform the following steps in the SCCM admin console before creating your SCCM ISO:

1. Import a Windows-based WIM file for deployment in the **Operating System Images** section.
2. Import an **Operating System Installer**.
3. Create an **Application** to install the vRealize Automation Agent. You can simply use the batch script and associated files from the previous section to create the application.

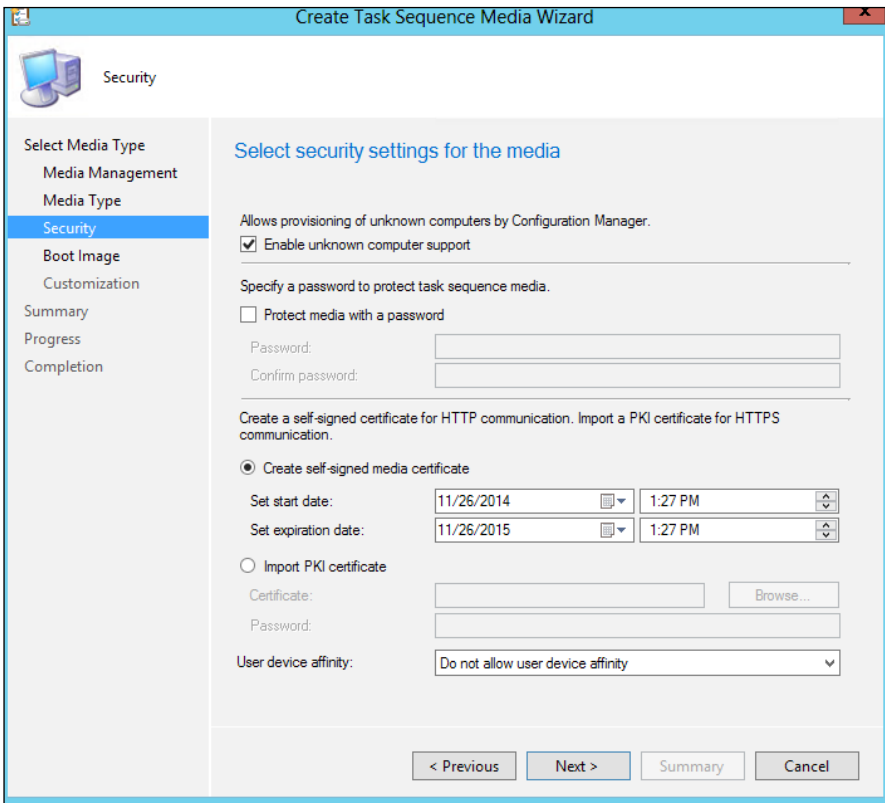
4. Create a **Driver Package** for VMware Tools and drivers.
5. Import the **VMware Tools and Drivers**. They can be found on an existing Windows VM in the C:\Program Files\Common Files\VMware\Drivers directory.
6. Add the **VMware Tools** drivers to the **Boot Image**.
7. Create a **Task Sequence** to install Windows and deploy it to the appropriate device collection in SCCM. For our example, we are using the collection titled **Deployments**. The task sequence should also deploy the vRealize Automation agent. You can use the script in the MDT section to create a SCCM package to complete this requirement.
8. The task sequence must be deployed with an **Action of Install** and a **Purpose of Required**. Additionally, you must make the task available to **Configuration Clients, Media**, and **PXE**. Failure to do so will yield a boot loop on the VM you try to provision.
9. Update the **Distribution Point(s)** with these changes.

Now it is time to create our SCCM ISO. The easiest way to accomplish this task is to perform the following steps:

1. Login to the SCCM admin console and click on **Software Library** in the bottom left-hand corner.
2. Expand **Operating Systems** in the upper-left hand corner of the screen. Right click on **Task Sequences** and select **Create Task Sequence Media**, as shown in the following screenshot:



3. Select **Bootable Media**. Be certain to place a check in the box at the bottom of the screen that says **Allow Unattended Operating System Deployment**. If you fail to do this step, the ISO we are creating will prompt you for input, thus defeating the purpose of unattended VM provisioning. Hit **Next**.
4. Select whichever option is appropriate on the **Select how media finds the management point** screen. If you are in a primary site, or have a single server instance of SCCM, choose **Site-based Media** and hit **Next**.
5. Select **CD/DVD Set** and choose where you would like to store the ISO locally on the SCCM server. Remember the location because you will need to upload the ISO to the vCenter datastore. Then hit **Next**.
6. Be sure to remove the check mark beside **Protect media with a password**. Failure to do so will prompt you for input, which we are trying to avoid. Take the other defaults and hit **Next**, as shown in the following screenshot:



7. Now, you will need to select **Boot Image | Distribution Point | Management Point**. Then hit **Next**.
8. At this point, you could add additional tasks, but we will leave this as default. Hit **Next**. This will take you to **Summary** Screen. Hit **Next** and the ISO creation will begin.

This will take quite a while to generate and can hang at processing 0 percent for several minutes. When it's completed, upload the ISO to a shared storage device in your vCenter environment. For our example, we will be saving the ISO on a LUN titled `shared_storage` in a folder called `ISO`. Take a look at the minimum custom properties you will have to define to get this to work:

Properties (12)		New Property		
Name	Value	Encrypted	Prompt User	
Image.ISO.Location	shared_storage	No	No	
Image.ISO.Name	/ISO/SCCM_WinPE_x64.iso	No	No	
SCCM.Collection.Name	Deployments	No	No	
SCCM.Server.Name	sccm.domain.local	No	No	
SCCM.Server.Password	*****	Yes	No	
SCCM.Server.SiteCode	ACM	No	No	
SCCM.Server.UserName	sccmadmin@domain.local	No	No	
VirtualMachine.FloppyDrive.Attach	False	No	No	
VMware.Floppy.Drive.Mount	False	No	No	
VMware.Network.Type	VMXNET3	No	No	
VMware.SCSI.Type	lsiLogicSas	No	No	
VMware.VirtualCenter.OperatingSystem	windows8Server64Guest	No	No	

Under the **Build Information** tab of your WIM Blueprint, you will select the **Action Create**. For **Provisioning Workflow**, select **VirtualScmProvisioningWorkflow** from the drop-down list.



Be certain to save your Blueprint; publish and test it before providing it to your end users.

## Creating a Kickstart Blueprint

Kickstart is an installation method used in enterprises to deploy unattended Linux operating systems across the network. Like SCCM, Kickstart allows for scripts and commands to be injected during the post installation process to give a more complete end result, and involves no human interaction.



Since Kickstart is very common in enterprise environments, it makes sense to have vRA leverage the existing technology. At a high level, a Kickstart Blueprint will work in this fashion:

- A Linux ISO is mounted and boots a small kernel
- This launches several scripts. These scripts request a DHCP address and tries to connect back to a server that hosts the Linux media. It performs tasks you have defined, such as installing the software, third- party packages, and custom configurations.

Before we create the Blueprint, we need to update the Linux boot image and create an ISO. There is plenty of information available on setting up a Kickstart environment, and we are not going to go into the details of doing so here (In fact, you can co-locate your Kickstart scripts on your SCCM, WDS, or MDT server). The easiest way to create the ISO is to perform the following steps from a Linux workstation. The following steps should work for any distribution, but in this example, we are referencing RHEL6. Also note, the references to **ks** and **wget** are referencing the FTP protocol. However, you can use HTTP as well.

1. Mount your Linux distribution and copy the `isolinux` directory to a local directory. For this example, use `localdirectory`.
2. Edit your `isolinux.cfg` file to point to your Kickstart server. You should edit or append the line `ks=ftp://yourkickstartserver/ks.cfg`. Below is a copy of the `isolinux.cfg` file. Highlighted are the changes need to be made:

```
default vesamenu.c32
#prompt 1
timeout 10
display boot.msg
menu background splash.jpg
menu title Welcome to Red Hat Enterprise Linux 6.2!
menu color border 0 #ffffffff #00000000
menu color sel 7 #ffffffff #ff000000
menu color title 0 #ffffffff #00000000
menu color tabmsg 0 #ffffffff #00000000
menu color unsel 0 #ffffffff #00000000
menu color hotssel 0 #ff000000 #ffffffff
menu color hotkey 7 #ffffffff #ff000000
menu color scrollbar 0 #ffffffff #00000000
label linux
    menu label ^Install or upgrade an existing system
```

```

menu default
kernel vmlinuz
append initrd=initrd.img --device=eth0 --bootproto=dhcp
ks=ftp://yourkickstartserver/RHEL6/ks.cfg

label vesa
menu label Install system with ^basic video driver
kernel vmlinuz
append initrd=initrd.img xdriver=vesa nomodeset

label rescue
menu label ^Rescue installed system
kernel vmlinuz
append initrd=initrd.img rescue

label local
menu label Boot from ^local drive
localboot 0xffff

label memtest86
menu label ^Memory test
kernel memtest
append -

```

3. Save the config file and launch the following command to create an ISO, making sure that you run it from the working directory containing the `isolinux` folder you copied:

```

genisoimage -r -T -J -V "RHEL6AMD64" -b
isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -
boot-load-size 4 -boot-info-table -o
/localdirectory/Kickstart.iso /localdirectory/

```

4. When this is completed, upload the ISO to a shared storage device in your vCenter environment. For our example, we will be saving the ISO on a LUN titled `shared_storage` in a folder named `ISO`.
5. We will also have to make some changes to the `ks.cfg` script on your Kickstart server. In the `%post` section of the Kickstart config script, add the following line:

```

%post
function process() {
    while true; do
        /usr/bin/gugent --host=vRealize Automationiaas.domain.local
        --ssl --config=/usr/share/gugent/gugent.properties --script=/usr/
        share/gugent/site
    done
}











```

```
if [ $? -eq 0 ]; then
    break
fi
sleep 30
done
}
cd /tmp
wget ftp://yourkickstartserver/RHEL6/Packages/gugent-6.1.0-073014.x86_64.rpm
rpm -ivh /tmp/gugent-6.1.0-073014.x86_64.rpm
export AXIS2C_HOME=axis2
export PYTHONPATH=/usr/share/gugent/site/dops
pushd /usr/share/gugent
echo | openssl s_client -connect vRealizeAutomationiaas.domain.local:443 | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > cert.pem
process # SetupOS
process # CustomizeOS
popd
```

6. Save your `ks.cfg` file. If you plan on doing this on a Windows-based computer (not recommended) be sure to download and run `dos2unix.exe` so that the file can be interpreted correctly by the Linux installation. It's a free program and takes out all the hidden characters a Windows-based text editor adds to a file.
7. Let's take a moment to explain what is going on in the `%post` section of the `ks.cfg` script. We are performing the following steps:

1. Downloading the vRA Guest Agent. This is available at `https://vra.domain.local:5480/installer`. You will need to extract its contents and pull out the RPM appropriate for your distribution. The RPM file needs to be copied to the `Packages` folder on your Kickstart server. In our preceding example, this is:  
`ftp://yourkickstartserver/RHEL6/Packages`
2. We are using the `wget` command to grab the vRA RPM from the Kickstart server and installing the vRA Guest Agent RPM.
3. We are setting the OS to trust the vRA IaaS server's certificate.
4. Finally, we are signaling to the vRA IaaS server that the provisioning of the VM is complete by the `process # SetupOS` and `process # CustomizeOS` lines.

8. Now that we have our Linux based prerequisites in place, let's create a Blueprint for Kickstart. You should already know how to accomplish this task by now. The following screenshot shows you the specific custom properties you need to define:

Properties (5) <span style="float: right;">+ New Property</span>				
	Name	Value	Encrypted	Prompt User
 	Image.ISO.Location	shared_storage	No	No
 	Image.ISO.Name	/ISO/Kickstart.iso	No	No
 	Machine.SSH	True	No	No
 	VMware.Network.Type	VMXNET3	No	No
 	VMware.VirtualCenter.OperatingSystem	rhel6_64Guest	No	No



Note, using the VMXNET3 network card is a VMware Best Practice. As of RHEL6, the supported driver is built in the kernel. However, other distributions may vary. You could remove this property and use the E1000 driver that is supported by all the distributions or update the `Kickstart.iso` that we've created with the VMXNET3 drivers.

9. Under the **Build Information** tab of your Kickstart Blueprint, you will select the **Action** *Create*. For **Provisioning Workflow**, select **LinuxKickstartWorkflow** from the drop-down list.

Be certain to save your Blueprint; publish and test it before providing it to your end users.

## Multi-machine Blueprints

Multi-machine Blueprints are great for scenarios where you may need to deploy machines in different locations, such as on-premises and in the Cloud.

To create a multi-machine Blueprint, perform the following steps:

1. Click on the **Infrastructure** tab, then on **Blueprints**. Click on **Blueprints** once more on the left-hand side, and the center of your screen should be populated with the Blueprints we created in the previous steps.
2. Hover your mouse over the **New Blueprint** button, and then select **Multi-Machine**. You will notice that the **Blueprint Information** tab looks identical to the previous Blueprints we have already covered.

3. Fill out the information as discussed in previous examples. When complete, click on the **Build Information** tab.
4. This screen will appear different from previous examples. It is in this section that you can select multiple Blueprints that you have already created and add them into a single multi-machine Blueprint.
5. Start by selecting the **Add Blueprint** button.
6. Browse and select the existing Blueprints you want to add. Hit **OK**. Since you have probably created **Build Profiles** and are already associating them with existing Blueprints (if you haven't, shame, shame!), there is no need to visit the **Network, Scripting, or Properties** tab to complete this Blueprint.

However, you can explore and leverage them as you get more experience using multi-machine Blueprints.

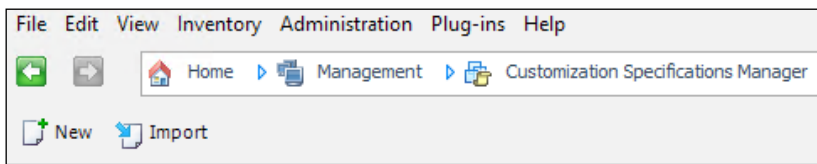
## Provisioning physical machines

vRealize Automation includes the support for physical machine provisioning, with configuration for iDrac and iLO (Dell and HP's respective remote management controllers). Essentially, vRA issues a command to wake up the physical machines and then gets the ISO needed to interact the vRA Blueprints via a PXE boot. However, we are not going to provide examples for the setup in this book.

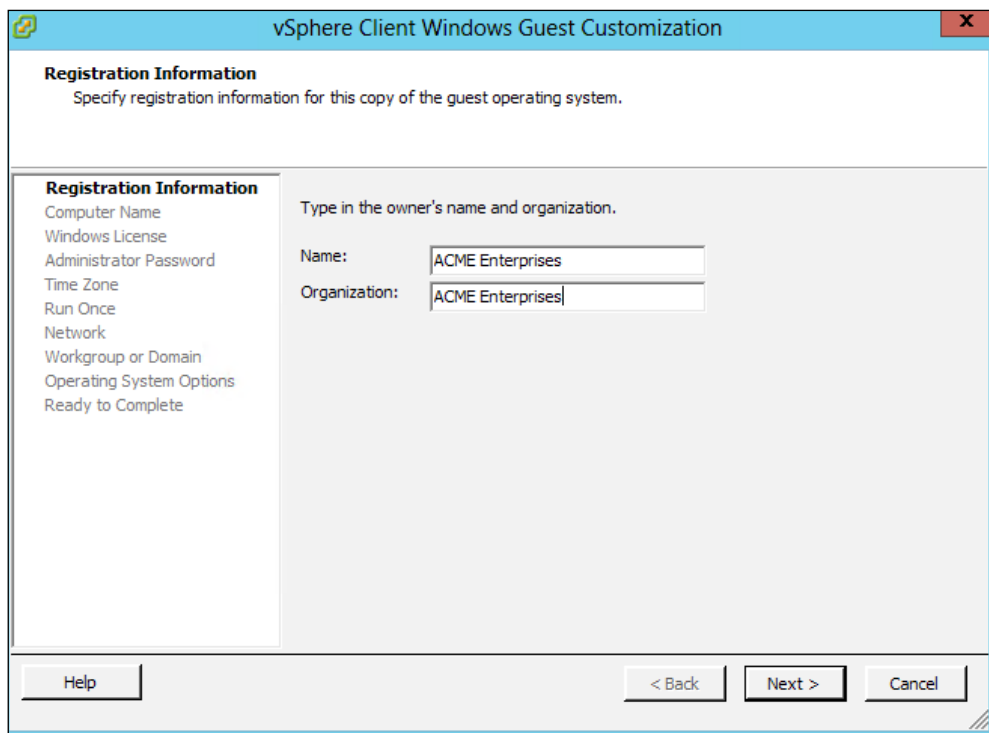
## Property dictionary

Property Dictionary settings are a great way to provide more interactive options for the end user during the provisioning process. For example, if we would like the end user to select which domain to join with their new VM, we can configure this option with **Property Dictionary**. Please note that the following example can only be used with clone and linked clone Blueprints. Perform the following steps:

1. Login to vCenter with the vSphere client. Navigate to the following location and click on **New**. This will launch a dialog box, where you can define the basics of your VM:



- Fill in all the fields that match your environment, as shown in the following screenshot:



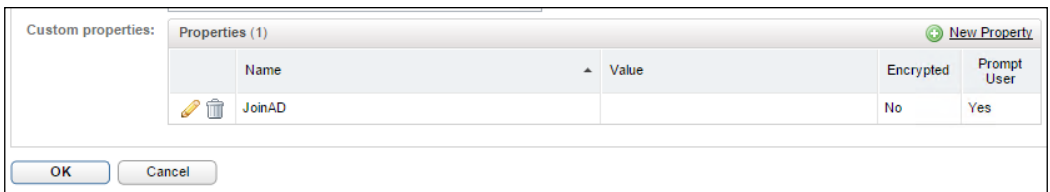
- Save your work and give it a name. For this example, we want to prompt the end user to join the domain. Therefore, we will name this customization `domain.local`.
- Back in the vRealize Automation web portal, go to **Infrastructure | Blueprints | Property Dictionary**. Click on **New Property Definition** in the upper right-hand corner. You will be required to fill in values for **Name**, **Display Name**, **Description**, **Control Type**, and **Required**. Fill in the values to match the following screenshot and click on the green check icon to save your settings:

Property Dictionary						
Manage the property dictionary.						
Property Definitions (2)						<a href="#">New Property Definition</a>
Name	Display Name	Description	Control Type	Required	Property Attributes	
JoinAD	Join AD	Join AD	DropDownList	Yes	<a href="#">Edit</a>	

- Click on **Edit** to define property attributes for this definition. Define the properties as displayed in the following screenshot. Click on the green check icon to save the changes:



- To associate the **Property Dictionary** item in the preceding example, edit the linked clone Blueprint we created earlier in this chapter. Navigate to the **Properties** tab and create a new custom property. Name it `JoinAD` and leave **Value** empty. Check the **Prompt User** option and save the changes, as shown in the following screenshot:



- When a user goes to the catalog and requests the VM to be deployed, the user will have to select to join the domain.

You can add additional vCenter customizations in the same property dictionary by comma separating the values. In the preceding example, if we wanted to provide the user different domain options, they could all be specified, comma separated, and would appear to the end user in the drop-down list we've created. We would also have to create additional Guest Customizations in vCenter, just like we did at the beginning of the example.

## Destroy a provisioned machine

Sometimes in the provisioning process VMs may get stuck or error out. While you could delete them directly from vCenter, this is not the best practice. Go to **Infrastructure | Machines | Manage Machines**.

From this screen, you can delete any VMs in the purview of vRA.

vRealize Automation deletes failed VMs after a period of time and makes troubleshooting difficult. Be sure to read *Chapter 11, Troubleshooting vRealize Automation 6.2*, where we illustrate how to disable this feature so you can analyze why the VM provisioning failed.

## Summary

We took a deep dive into Blueprints, illustrating examples of what is needed to get all of the ones included out-of-the-box up and running. We covered how to leverage unattended systems that already exist in our environment with vRA.

In addition to the Blueprints, we provided the essential custom properties that are required to get the provisioning portions to succeed. Without the right combination of custom properties, a provisioning request will fail. Unfortunately, the error messages are typically basic, which makes diagnosing the source of the error very difficult. We will cover these topics later.

In *Chapter 6, Creating Approval Policies* we will focus on creating approval policies.





# 6

## Creating Approval Policies

In this chapter, we are going to focus on **Approval Policies** and how they can assist in the control and governance of our growing vRealize Automation 6.2 environment. We have installed, designed, and now implemented vRA. We have also created blueprints, which are the instruction sets necessary to provision machines.

When discussing blueprints, we mentioned how we would provide a baseline for compute. Within that discussion, we defined maximum compute values as well. This is to make sure we define a hard limit on the resources a machine can consume after being deployed from a blueprint.

What happens if an end user wants a set of compute resources somewhere in between these ranges?

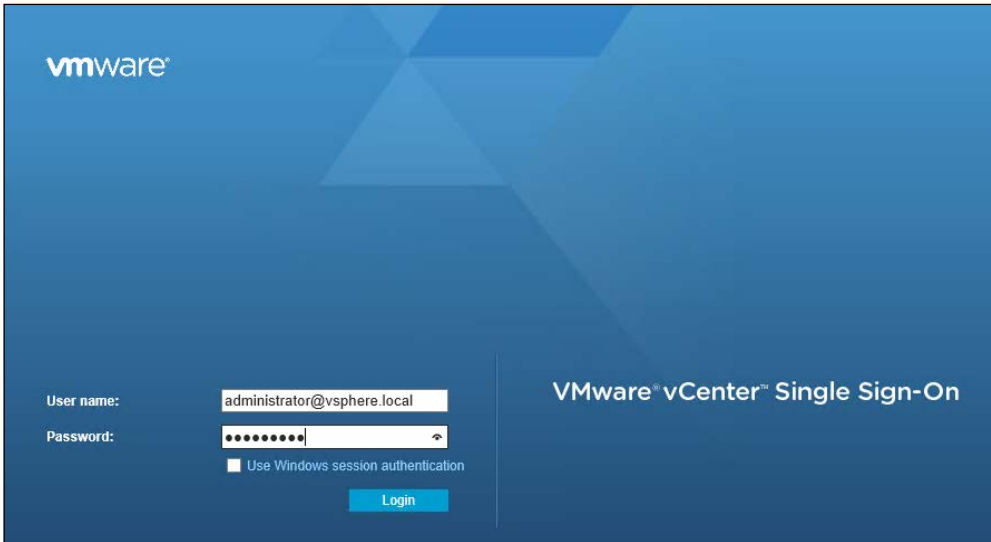
The solution is Approval Policies. If a user wants something, all they have to do is ask (virtually speaking). After all, we discussed in the pilot phase that we need to take a user-centric approach when we define the success factors that are involved in implementing vRA. It only makes sense that we would allow the users to customize their requests to fit their needs. In this chapter, we will be covering the following topics:

- Getting started with Approval Policies
- Creating a Post Approval Policy

### Getting started with Approval Policies

Since we have a basic understanding of what Approval Policies can do for us, let's build a couple of them. For starters, let's create an Approval Policy for a request that is for more than one vCPU.

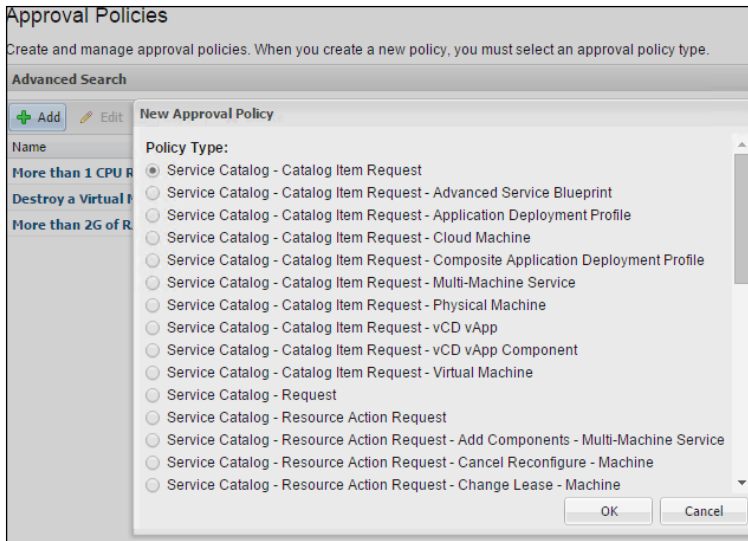
Open a web browser and head to `https://vRA.domain.local/vcac`. You will see a screen similar to this one:



Login with the domain-based vRA administrator account you have been using to create the examples in the book.

Click on the **Administration** tab, and then click on **Approval Policies**.

Click on the **Add** button and you will see this:



We are going to select **Service Catalog | Catalog Item Request | Virtual Machine**. Click on **OK** to proceed. The following screen will appear:

The screenshot shows a configuration window for an approval policy. The 'Name' field is filled with 'More than 1 CPU Requires Approval'. The 'Description' field contains 'Any Machine Request for more than 1 CPU requires Approval'. The 'Policy Type' is set to 'Service Catalog - Catalog Item Request - Virtual Machine'. The 'Status' is 'Active'. The 'Last Updated On' date is '11/5/14'. There are two tabs: 'Pre Approval' and 'Post Approval'. At the bottom, there is a 'Levels' section with a plus sign icon.

Provide a **Name** for the Approval Policy, such as *More than 1 CPU Requires Approval*, as shown in preceding example. Be sure that you set the **Status** to **Active**.


Under the **Pre Approval** tab, click on the green plus sign next to **Levels**.

Now as shown in the following screenshot, provide a name in the **Name** field and select **Required based on conditions**, as shown here:

The screenshot shows the 'Add Level' dialog box. The 'Name' field contains 'Check Number of CPUs'. The 'Description' field is empty. The 'Is manual approval required?' section has 'Required based on conditions' selected. The condition is 'CPUs > 1'. The 'Who are the Approvers?' section has 'Specific Users and Groups' selected. The 'Business group' dropdown is open, showing 'CPUs' selected. The 'Add' and 'Cancel' buttons are at the bottom right.

We are going to choose **CPUs** under the **Business Group** category. We will then select the **>** (greater than) symbol and provide a value of 1. This statement is basic, but it means "*check to see if the request is being made for more than one CPU*". If the condition is met, what will happen? We are going to continue this exercise by specifying a person who will approve the request.

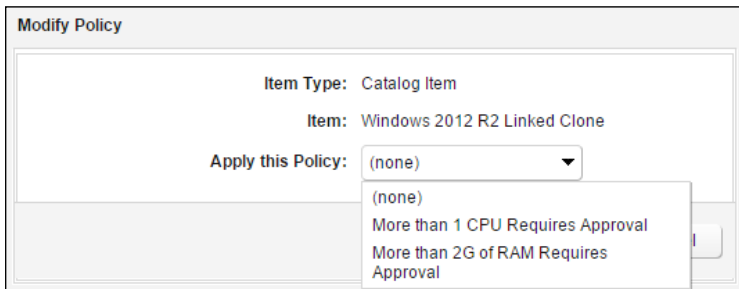
On the right-hand side of the screen, under **Who are the Approvers?** search for a domain user (such as the one you are using to create this request) to be the approver. Click on the **Add** button.

 Make sure the domain account you are going to use as an approver has a valid, routable e-mail address associated with it. You can set this parameter by opening the **Active Directory Users and Computers** snap-in and then insert the e-mail address in the **E-mail** field associated with the account. This is necessary when a user may be using an external e-mail address, such as Gmail. Fortunately, even if you provide an invalid e-mail address, the approval will still appear on the vRA web portal.

Let's take a moment to cover the flexibility of adding multiple approvers to the Approval Policy. Let's say we need the manager of the software developer group to approve creating the VM, as well as the manager of the IT team. We could require both of them to approve the VM request. If one of the two managers rejects the request, the VM is not provisioned. Based on the business needs, you can set up numerous combinations to govern the Approval Policies you create.

Once we have saved our first Approval Policy, we have to tie it to an existing entitlement. In order to do so, go to **Catalog Items**, then click on **Entitlements**. Select the entitlement you wish to modify (software developers, anyone?).

Click on the **Items and Approvals** tab. Under **Entitled Catalog Items**, click on the drop-down arrow next to the item you wish to link to an Approval Policy. Let's link this to the **Windows 2012 R2 Linked Clone** blueprint. Click on **Modify Policy**, as shown here:



Select the Approval Policy you've created and click on **OK**. At the bottom of the screen, click on **Update**.

You have successfully created an Approval Policy. If an end user from the software development team requests a Windows 2012 R2 VM with more than one CPU, vRA will send you an e-mail, as shown here:

[Reply](#) [Reply All](#) [Forward](#)

Fri 11/14/2014 6:38 PM

vRA@domain.local

**Action Needed: Request for Windows 2008 R2 Clone from Template**

To IT Manager

---

### Request Information

**Description**

**Reasons**

---

### Machine Details

**Lease duration**  
Indefinite

**CPUs**  
3

**Memory**  
4,096 MB

**Disk 1 size**  
40 GB

---

[Click here to view the details](#)

---

Click one of the following choices to respond to this email. Do not edit the generated subject line.

- [Approve](#)
- [Reject](#)

The user request will remain in a pending state until you approve or reject it. You can approve or reject it by clicking on the link, which is in the body of the approval e-mail. If you happen to be in the vRA portal, you can click on the **Inbox** tab, and the approval request will be listed.

If the necessary approvals are in place, the VM will be provisioned. If they're rejected, the VM will never be created.

We can build more options into the Approval Policy as well. We could easily add a second line that would require approval if we requested 4 Gig of RAM instead of the basic 512 Meg that we used when we built our blueprints. Can you add this on your own?

## Creating a Post Approval Policy

After a VM has been provisioned and the end user has worked in it for a few days, they notice they need more CPU. How can we allow them to request the compute resource they need now that the VM is in service? That is a great question. Let's edit our previous Approval Policy and instead of adding conditions to the **Pre Approval** tab, let's utilize the **Post Approval** one.



In order to edit an existing Approval Policy, you need to remove it from being linked to any existing entitlements. If you do not perform this step, you will be unable to add new conditions to an existing policy.

Under the **Post Approval** setting, add a condition that sends out an approval request if the user requests more than 4 Gig of RAM. Since we already have the CPU and memory hot-add function enabled on our VMs, as soon as the request is approved, the compute resource will be added dynamically.

## Summary

Approval Policies are designed to allow vRealize Automation administrators to control the resources they provide to end users. We covered a couple of scenarios and examples of Approval Policies and the steps necessary to create them. Approval Policies can be created before or after a VM or service has been provisioned.

In *Chapter 7, Installing and Configuring vRealize Automation Application Services 6.2*, we will set up the vRealize Automation Application Services, which will allow us to provision applications through the vRA web portal.

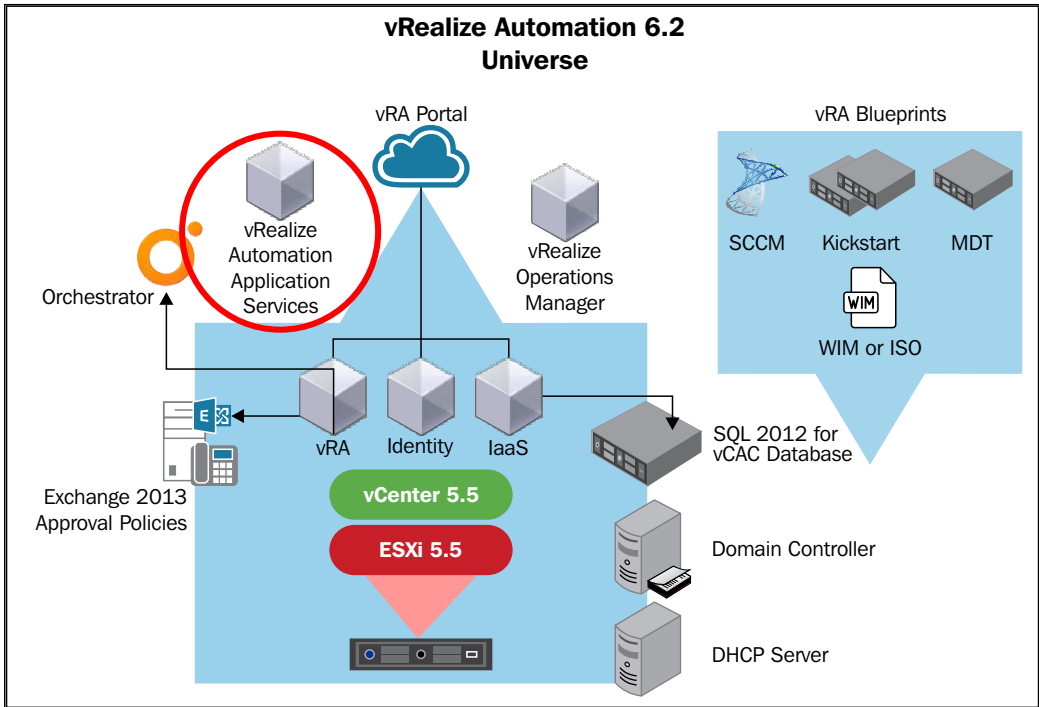
# 7

## Installing and Configuring vRealize Automation Application Services 6.2

In this chapter, we are going to get vRealize Automation Application Services installed and configured in our environment. (vRealize Automation Application Services was formally known as **Application Director (AppD)**). You may wonder why this was not included in an earlier chapter of this book. Well, wonder no more. Quite simply, not only do you have to have the Identity Appliance, IaaS server, and vRA appliance installed and configured, you also have to have a wealth of web portal components configured, as well an understanding of how they work. This warranted a few chapters on Blueprints and Approval Policies first.



Now that we have these foundations and concepts in place, we can introduce a new component to our vRA universe. Take a look at the highlighted component of the following diagram to see where the vRealize Automation Application Services Appliance (AppD) fits in:



vRealize Automation Application Services automates application provisioning. This includes deploying, configuring, and updating the applications in your environment. Using AppD helps simplify complex deployments of customized applications in your environment. For example, you can customize the deployment of a SQL server, ensuring that it is installed on the specific partitions and locations that your environment requires.

We will be covering the following topics in this chapter:

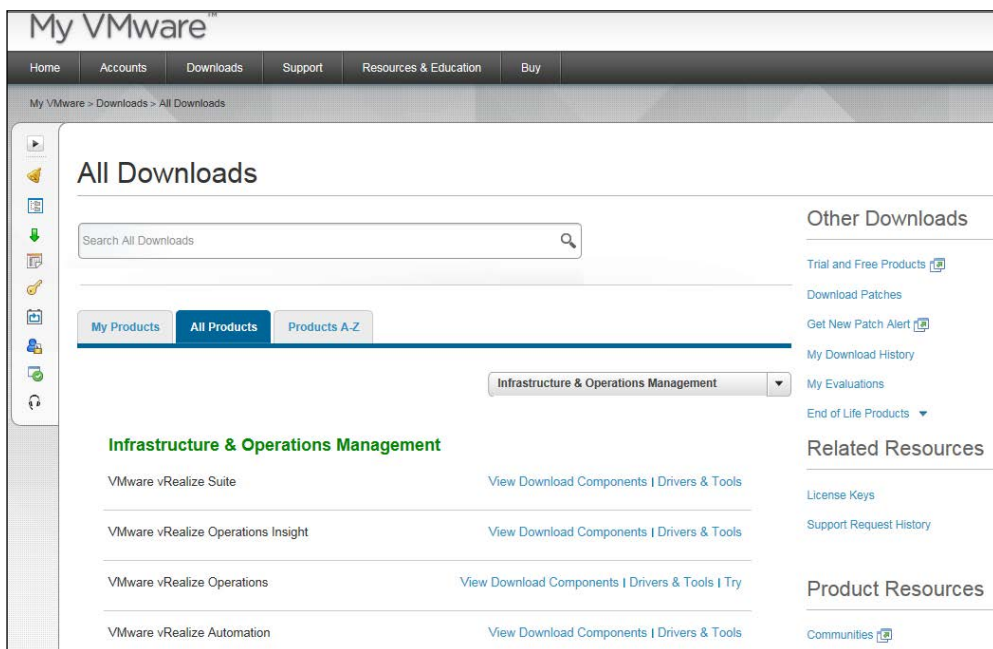
- Downloading the vRealize Automation Application Services 6.2 software
- Installing the vRealize Automation Application Services Appliance
- Configuring the vRealize Automation Application Services Appliance
- Configuring the AppD web portal

- Installing the AppD Bootstrap agent on a VM
- Configuring SQL for provisioning

## Downloading the vRealize Automation Application Services 6.2 software

To download the vRealize Automation Application Services 6.2 software, you must first go to [www.vmware.com](http://www.vmware.com) and login to the **My VMware** portal. If you do not have an account, you can sign up at no charge.

Once you've logged in, click on the **All Downloads** icon on the left-hand side of the portal, then click on the **All Products** tab. Under **Infrastructure & Operations Management**, click on **View Download Components**, which is to the right of **VMware vRealize Automation**. Take a look at the screenshot for an example:



You will see the different versions available for download. We will download the Enterprise edition of vRealize Automation Application Services 6.2. Download the appliance to a shared folder in your lab environment.

# Installing the vRealize Automation Application Services Appliance

Follow these steps to install vRealize Automation Application Appliance:

1. Open vSphere and login to your vCenter instance. Note that we will be using the vSphere *traditional* GUI instead of the web-based version.
2. Select **File** in the upper left-hand corner of vSphere and select the **Deploy OVF Template** option.
3. Browse to the location where you downloaded the software from in the previous section and select the `vRealizeApplication Services OVA` file. Click **Next**.
4. The next screen will list **Product** (product name), **Version**, **Vendor**, **Publisher**, **Download size**, **Size on disk**, and **Description**. Click **Next**.
5. Accept the VMware license on the next screen and click **Next**.
6. Accept the default name and select the install location from the **Name and Location** option. Click **Next**.
7. Select the cluster where you want to deploy the Appliance and click **Next**.
8. Select the appropriate resource pool from the **Resource Pool** option and click **Next**.
9. Select what storage location you would like to have the appliance installed in from the **Storage** option and click **Next**.
10. Select the disk format from the **Disk Format** option. By default, it is selected for Thick Provisioning. However, you can change this to Thin Provisioning format, if you wish. Make your choice and click **Next**.
11. Under **Network Mapping**, select the network appropriate selection and click **Next**. Remember, we keep all the machines on the same subnet for the examples used in this book. Click **Next**.
12. You will now be prompted for the **IP Allocation Policy**. It is recommended that you choose **Fixed**. This will represent a static IP. Click **Next**.
13. Under the **Properties** option, enter a static IP. It is recommended to keep this IP on the same subnet as your other vRA components.
14. Once complete, click **Next**. Place a check mark in the box that says **Power on after deployment**, then click **Finish**.

In **Domain Name System (DNS)**, make a static entry for your vRA Application Services Appliance. In this exercise, we will use `vRAappd.domain.local` as the **Fully Qualified Domain Name (FQDN)**.

# Configuring the vRealize Automation Application Services Appliance

Follow these steps to configure the vRealize Automation Application Services:

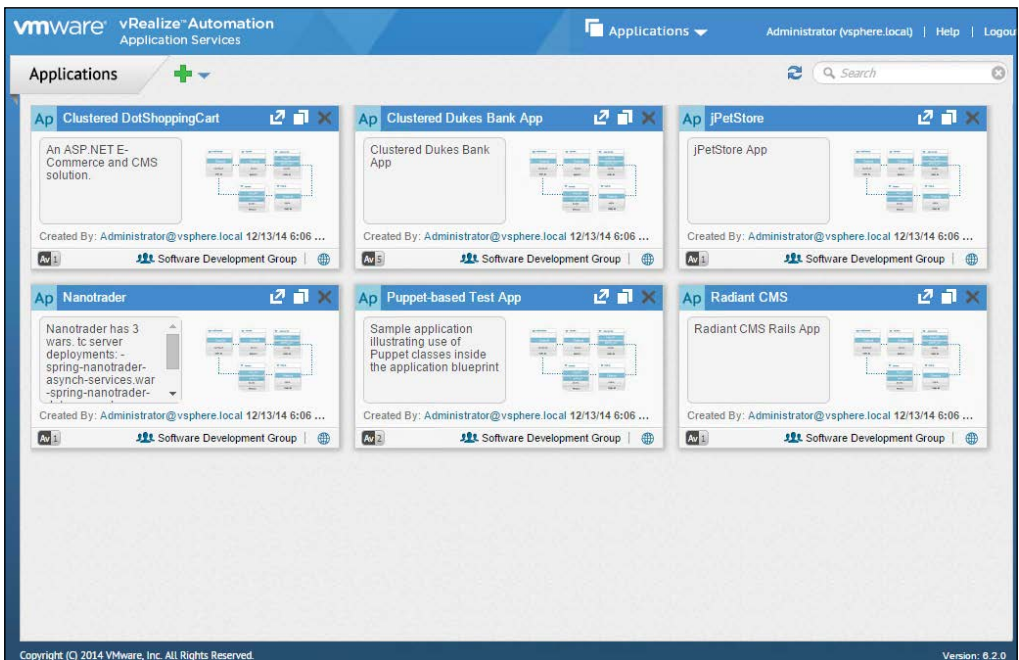
1. Open the console of the AppD appliance and enter your vRA license key. This is the same key you used on the vRA virtual appliance and the web portal. Don't forget to include the dashes.
2. Once it is registered successfully, you will be prompted to set the password for the OS root account. Enter a password then hit *Enter*.
3. You will be prompted for a second password. This is for `darwin_user`. You will have to use this account to *SSH* into the appliance, if needed. Set a password and hit *Enter* to continue.
4. You will now see the installation continue and the server start. This normally takes a few minutes to complete before you are prompted to provide more information. When you see **Do you wish to use this instance of Application Services for 6.0.1 migration?**, it is time to continue. Since this is a brand new installation, we will respond by hitting *N* and then *Enter*.
5. Next, you will enter the URL of the vRA appliance. Based on our running example, you would enter `https://vRA.domain.local`.
6. For system administrator username, enter `administrator@vsphere.local`. Hit *Enter*.
7. Enter the password for `administrator@vsphere.local` and hit *Enter*. This will start the registration process with vRA.



If at any time these steps fail, simply hit *Y* and *Enter* to repeat the process.

8. Once the registration is complete, you will be prompted to install out-of-the-box samples. It is recommended to respond with a *Y*, and then continue.
9. You will be prompted to enter a tenant name. You should enter `vsphere.local` and hit *Enter* to continue.
10. Next, you will be prompted for a username. This should be the name of a tenant administrator in the format of `user@domain.local`. It is *NOT* recommended to use `administrator@vsphere.local`. Use the vRA administrator domain account we defined earlier.

11. Login to the vRA web portal and click on the **Administration** tab. Click on **Groups**. If you have not already defined one, click on **Add** to create one. You can browse through **Identity Stores** for a list of AD groups. Find the group you want and add all of the available permissions on the right-hand side. If the account you are using does not have all the permissions associated with it, the out-of-the-box samples will fail to import.
12. Enter the password for the username from the previous step and hit *Enter*.
13. Finally, you will be prompted to enter a name under the **Business Groups** tab. For our example, we have been using `Software Developers`. Since this group has spaces, be sure to enter it on the command line in quotes, such as `"Software Developers"`.
14. Once this is complete, you will be prompted to create a new password again. Do so and hit *Enter*.
15. The appliance will continue to load. Once it is up, be sure to set the appropriate time zone.
16. Once you have set the time zone, you can login to the AppD portal at `https://vRAappd.domain.local:8443/darwin/org/vsphere.local`.
17. Login with the same AD account you have been using to configure the vRA web portal. Use the `user@domain.local` format.
18. If your samples have installed correctly, this is how your screen should look:



If it is blank, the samples did not imported properly. Go through *Chapter 11, Troubleshooting vRealize Automation 6.2*, to find out how to resolve this issue.

## Configuring the AppD web portal

Although we have loaded the samples, we still have to perform some configurations before we can start using AppD. Follow these instructions to configure AppD, and tie it into vRA:

1. Open a web browser on a machine that is on the same subnet as the vRA web portal. This ensures there are no firewalls in place. In our example, this would be the following URL: `https://vRAappd.domain.local:8443/darwin/org/vsphere.local`.
2. Since this is the first time you are logging in, you will get a certificate warning. This is because it uses the certificate that comes with the default appliance and is not a *Trusted Publisher*. Continue until you get to the screen which shows a username and password prompt.
3. Login with the credentials you used in the previous section. Remember to do so in the `username@domain.local` format.
4. You will land on the **Applications** page. In the upper right-hand corner, just to the left of your username, select the drop-down arrow and select **Cloud Providers**. This is where we will add our vRA web portal. Click on the *green plus sign* to add a provider, as shown in the following screenshot:

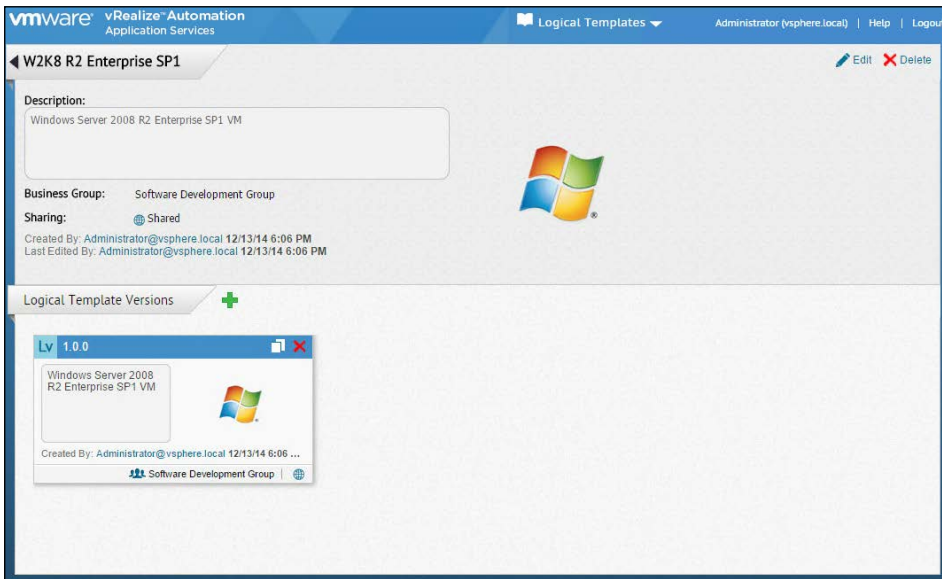
The screenshot shows the 'New Cloud Provider' configuration interface in VMware vRealize Automation. The form is titled 'New Cloud Provider' and includes the following fields and options:

- Name:** vRA
- Description:** enter description
- Cloud Provider Type:** vCAC
- vCAC Infrastructure IP/Host Name/URL:** https://vraiaas.domain.local
- User Name:** vra@domain.local
- Password:** masked with dots
- Business Group:** Software Development
- Sharing:** Private
- Validate Connection:** Disabled (red circle with slash)

At the bottom, there is a 'Templates' section with a table:

Name	Description

5. Fill out the fields as listed in the preceding screenshot. Be sure to click on **Validate Connection**.
6. Once this is completed, click on *plus sign* beside **Templates**. This will allow you to import all the Blueprints you've created in vRA up to this point. Be sure to save your settings by clicking on **Save**.
7. In the upper right-hand corner, just to the left of your username, select the drop-down arrow and select **Deployment Environments**. Click on the *green plus sign* to add an environment. Give the environment a Name and select the Cloud Provider you created in the previous step. Save your changes.
8. In the upper right-hand corner, to the left of your username, select the drop-down arrow and select **Logical Templates**. This will show a listing of all of the out-of-the-box samples, mainly comprised of operating systems. You must edit these samples before they can be used. Additionally, these samples can only be used with **Linked Clone** or **Clone Blueprints**.
9. There are no samples included with Windows 2012 R2. Therefore, you will have to create a Linked Clone Blueprint of Windows 2008 R2 to use these samples.
10. Once a Windows 2008 R2 Clone Blueprint is created, you will need to edit the **W2K8 R2 Enterprise SP1** sample. Click on it, as shown in the following screenshot:



11. Click on the window displayed under **Logical Template Versions**. It will take you to the following screen:

W2K8 R2 Enterprise SP1 v1.0.0

Name: W2K8 R2 Enterprise SP1    Version: 1.0.0    Tags: OS Templates

Description: Windows Server 2008 R2 Enterprise SP1 VM

Business Group: Windows Users    Sharing: Shared

Cloud Template Mapping

Cloud Provider Name	Cloud Template
vRAC	Windows 2008 R2 Clone from Template

Services Included

Service Name	Description
No items exist.	

12. Click on **Edit** in the upper right-hand corner and then hit the *plus sign* under the **Cloud Template Mapping**. This is where you will add the Windows 2008 R2 Clone Blueprint you created. Save your work. Feel free to repeat this process for the other samples you might use and were imported when setting up AppD.

Congratulations! You have setup the prerequisites for using AppD. Proceed to the next section which details Application Provisioning for SQL.

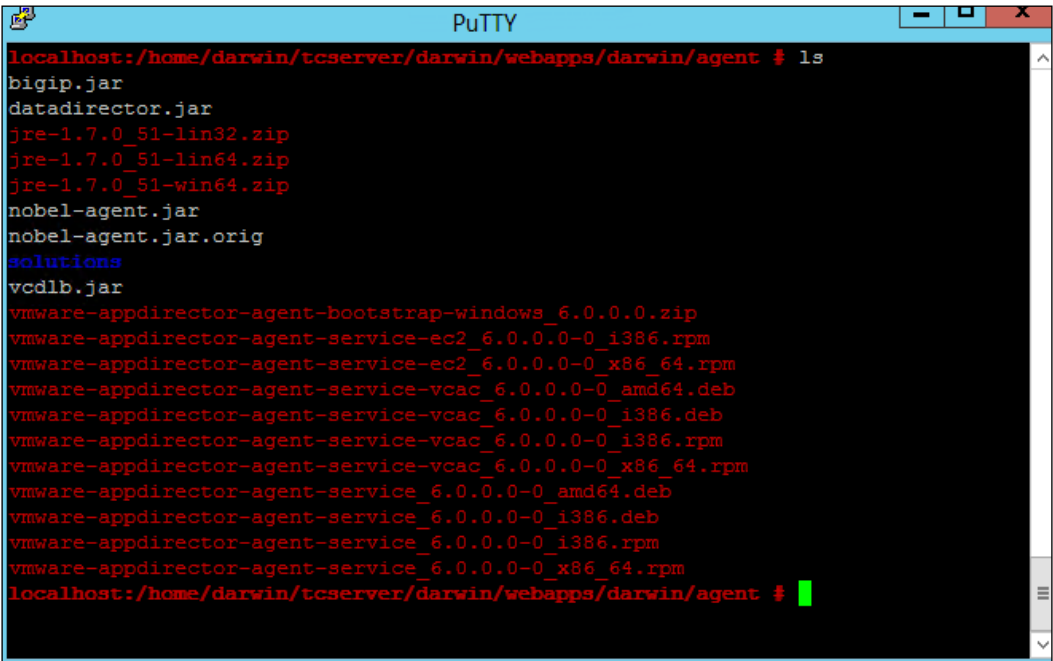
## Installing the AppD Bootstrap agent on a VM

Just like the agents for vRA, we have an agent for AppD as well. Without this agent, the application or services from AppD do not get deployed correctly. You can install the agent directly to a VM, and save it as a template, or install it as part of an automated deployment process, such as Kickstart, MDT, or SCCM. You must have VMware tools installed prior to performing the AppD Bootstrap agent installation. We will continue with the following examples using Windows 2012 R2 as our guest VM.

1. First, we need to gather some prerequisites. For this, we will need the following:
  - The Java JRE package that comes with AppD. You can download at [https://vRAappd.domain.local:8433/darwin/agent/jre-1.7.0\\_72-win64.zip](https://vRAappd.domain.local:8433/darwin/agent/jre-1.7.0_72-win64.zip)



- The AppD agent can be found at [https://vRAappd.domain.local:8433/darwin/agent/vmware-appdirector-agent-bootstrap-windows\\_6.0.0.0.zip](https://vRAappd.domain.local:8433/darwin/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip)
  - NTRights.exe, which is part of the **Windows Server 2003 Resource Kit Tools** can be found at <http://www.microsoft.com/en-us/download/details.aspx?id=17657>
2. For completeness, you can see all of the available Bootstrap agents for AppD by logging into the AppD server with SSH. Use `darwin_user`, and then `su` to become root. The Bootstrap agents appear in the `/home/darwin/tcserver/darwin/webapps/darwin/agent` directory, as shown in the following screenshot:



```
localhost:/home/darwin/tcserver/darwin/webapps/darwin/agent # ls
bigip.jar
datadirector.jar
jre-1.7.0_51-lin32.zip
jre-1.7.0_51-lin64.zip
jre-1.7.0_51-win64.zip
nobel-agent.jar
nobel-agent.jar.orig
solutions
vcldb.jar
vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip
vmware-appdirector-agent-service-ec2_6.0.0.0-0_i386.rpm
vmware-appdirector-agent-service-ec2_6.0.0.0-0_x86_64.rpm
vmware-appdirector-agent-service-vcac_6.0.0.0-0_amd64.deb
vmware-appdirector-agent-service-vcac_6.0.0.0-0_i386.deb
vmware-appdirector-agent-service-vcac_6.0.0.0-0_i386.rpm
vmware-appdirector-agent-service-vcac_6.0.0.0-0_x86_64.rpm
vmware-appdirector-agent-service_6.0.0.0-0_amd64.deb
vmware-appdirector-agent-service_6.0.0.0-0_i386.deb
vmware-appdirector-agent-service_6.0.0.0-0_i386.rpm
vmware-appdirector-agent-service_6.0.0.0-0_x86_64.rpm
localhost:/home/darwin/tcserver/darwin/webapps/darwin/agent #
```



Make sure there is no vRA guest agent installed on the VM or template. If it has been installed, you will have to manually uninstall the guest agent, stop the service and then run `sc` and delete vCAC guest agent service. Installing the AppD Bootstrap agent will configure the vRA agent as part of the installation process.

3. Extract the Java ZIP file to `C:\opt\vmware-jre` on the Windows VM. If you have **User Account Control (UAC)** enabled, you will need to right-click on each file that was part of the ZIP archive and click on the **Unblock** button.
4. Take the `NTRights.exe` file and place it in `C:\opt\`.
5. Download and extract **VRMGuestAgent**. This should be extracted directly to the `c:` drive.
6. Extract the AppD Bootstrap agent ZIP file to the `C:\opt` directory.
7. Next, open a Command Prompt and change directories so that you are in the same location where you extracted the AppD Bootstrap agent's files. Run the following command to install the agent:

```
install.bat password=Password cloudProvider=vcac
vcacServer=vRAiaas.domain.local httpsMode=true vcacServerPort=443
```

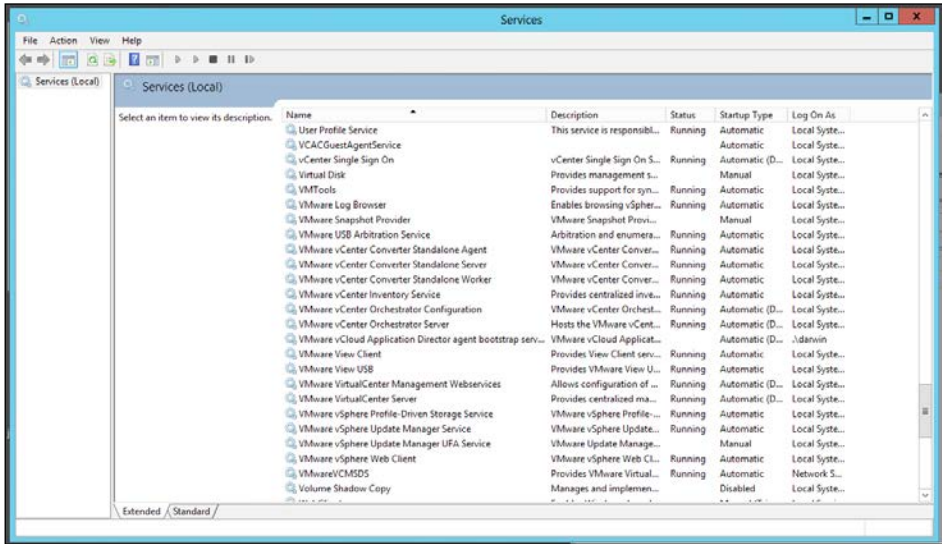


This command is case sensitive. Also note that `vcacServer` is referencing your IaaS server.

8. The preceding command will create a user on the local machine named `darwin` using the password we specified above. Make sure that the password you select meets the complexity requirements of your environment.
9. If this is completed successfully, you should get a screenshot similar to the following:

```
Administrator: cmd.exe
The Install phase completed successfully, and the Commit phase is beginning.
See the contents of the log file for the C:\opt\vmware-appdirector\agent-bootstrap
ap\bootstrapWin.exe assembly's progress.
The file is located at C:\opt\vmware-appdirector\agent-bootstrap\bootstrapWin.In
stallLog.
Committing assembly 'C:\opt\vmware-appdirector\agent-bootstrap\bootstrapWin.exe'
Affected parameters are:
  logtoconsole =
  assemblypath = C:\opt\vmware-appdirector\agent-bootstrap\bootstrapWin.exe
  logfile = C:\opt\vmware-appdirector\agent-bootstrap\bootstrapWin.InstallLog
The Commit phase completed successfully.
The transacted install has completed.
Add custom configuration to the service...
[SC] ChangeServiceConfig SUCCESS
Look for ntrights.exe...
[SC] ChangeServiceConfig SUCCESS
Granting SeServiceLogonRight to darwin ... successful
Agent bootstrap installation completed.
C:\opt\vmware-appdirector\agent-bootstrap>
```

10. Check the *services snap-in* and make sure you have two new services. One named **VCACGuestAgentService** and the other named **VMware vCloud Application Director agent bootstrap service**:



We can now save this template, or script this out in a fashion similar to our MDT or SCCM deployment.

## Configuring SQL for provisioning

1. Go back to the **Applications** page in AppD.
2. Click on the *green plus sign* in the upper left-hand corner and select **Create New Application**.
3. Give the application the name **SQL 2008**, select a Business Group (such as Software Developers), and make the **Sharing** option **Shared**. Hit **Save** in the upper right-hand corner.
4. Before going any further, we have to provide an **Application Version**. Click on the *plus sign*. You can accept the default value of **1.0.0**. Be sure to save your changes.
5. Now, click on **Create Blueprint**. You will notice the middle workspace is blank, but there are populated options to the left and right-hand side. Under **OS Templates**, drag and drop the bottom left-hand selection to the workspace. It's called **W2K8 R2 Enterprise SP1**.

- On the bottom right-hand side, under **Services**, drag **Microsoft SQL Server 2008 Express** directly on top of the W2K8 R2 Enterprise SP1 icon. Hit **Save** in the upper right-hand corner of the screen.



There are a lot of powerful options in this section and the reader is encouraged to experiment. We are going to keep them basic for this chapter and the examples used in this book.

- Once saved, hit the **Deploy** icon. A pop-up box will appear, prompting you to provide a new deployment profile name. Let's call it SQL 2008. Hit the **Deploy** button on the pop-up box to continue.
- You will be taken to a four-step deployment screen. Now, take a look at the following example:

Deployment Environment		
Name	Description	Cloud Provider
vRAC_vCenter		vRAC

Map Details

VM Templates		
Node Name	Logical Template	Cloud Template
W2K8_R2_Enterprise_SP1_with_SQL_Server_v1.0	W2K8 R2 Enterprise SP1 with SQL Server	Windows 2008 R2 Clone from Template

Networking		
Logical Network Name	Nodes Using this Network	Cloud Network Name
Default Logical Network	W2K8_R2_Enterprise_SP1_with_SQL_Server_v1.0	template specific

- Click on the **Map Details** button and make sure that there are no errors.
- Click on the **Next** button and take the default properties listed on the **Step 2: Application Properties** screen. Click on **Next**.
- On the **Step 3: Execution Plan** screen, again, take the defaults, as shown in the following screenshot:



12. On the **Step 4: Review** screen select **Publish** and in a few minutes, SQL 2008 will appear as a **Catalog Item** in the vRA portal.



Return to the vRA web portal and go to **Catalog Items**. Associate SQL 2008 with the appropriate service. Next, add the additional entitlements and actions to the group that will use this product. At this point, it is ready for provisioning.

You could also deploy the application from the AppD web portal. This is helpful for troubleshooting purposes, so you can see at what point in the Execution Plan the blueprint potentially fails. Also, note the AppD web portal will also keep a record of the deployments, even if you have published and deployed the service through the vRA web portal.

## Summary

We have spent several chapters concentrating on setting up Infrastructure as a Service. In this chapter, we begin to experience the next step in making our vRealize Automation environment with the introduction of Application Provisioning through AppD. AppD allows us to provide Applications as a Service to end users. We discussed how to install and configure the vRealize Automation Application Services appliance. We also imported out-of-the-box samples, and modified them so we could publish them to the vRA portal.

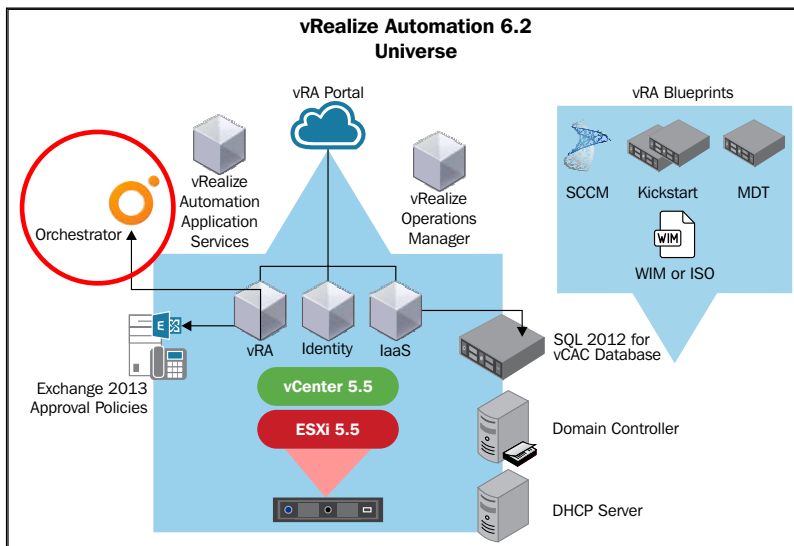
In *Chapter 8, REST API and vRealize Orchestrator*, we will focus on the REST API.

# 8

## REST API and vRealize Orchestrator

In this chapter, we are going to explore the REST API and show you how it pertains to vRealize Automation. We will use vRealize Orchestrator to leverage the APIs within vRA to perform additional, customized automation.

In the previous chapter, we introduced vRealize Automation Application Services and showed how it allows us to provide applications and services to end users through a web portal. Now that we have these foundations and understandings in place, we can introduce another layer of complexity to our vRA universe. Take a look at the highlighted component of the following diagram to see where the REST API fits in:



In this chapter, we will discuss the following topics:

- Introducing REST
- Application programming interface
- vRealize Orchestrator configuration
- Navigating through vRealize Orchestrator Client
- Defining our example customized workflow
- Configuring the prerequisites in vRO
- Creating a custom workflow with vRealize Orchestrator Client
- Testing a customized workflow
- Adding workflows to a vRA web portal
- Additional workflow tools

## Introducing REST

REST was developed by Roy Fielding at the University of California, Irvine and published in his doctoral dissertation in 2000. The goal of REST is to provide a method to design improvements and have these improvements systematically evaluated before they are deployed across the Internet. Think of REST as a litmus test that proves the extensions of components, connectors, and the data elements work prior to being deployed. The end goal is to reduce latency and network communication, while maximizing the independence and scalability of components on Internet deployed systems.

**REST** is short for **REpresentational State Transfer**. It is most commonly applied to the development of web services. When an item is referred to as **RESTful**, it means it conforms to the following architectural constraints:

- **Uniform interface:** This is the key element that separates itself from other network-based architectural styles. Essentially, it defines the interface between clients and servers. It separates the components of the architecture so that all elements can evolve independently. There are four guiding principles that make up the uniform interface:
  - **Resource identification:** Individual resources are identified using **URIs (Uniform Resource Identifiers)**. As an example, when a request for data is made from a web page, the server containing the web data (which is stored in a database) does not send the database itself, but something similar to XML or JSON that represents some of the database records.

- **Manipulation of resources through representations:** This means once a client holds a representation of a resource, it can modify or delete the resource on the server, provided the proper permissions have been granted on the server for these actions to take place.
  - **Self-descriptive messages:** Each client request and the associated server response is referred to as a message. In a RESTful application design, each message contains all the necessary descriptive information that is needed to complete a task. These messages typically contain the these HTTP-based actions: GET, HEAD, OPTIONS, PUT, POST, and DELETE.
  - **Hypermedia as the Engine of Application State:** The fourth and final component of the uniform interface is Hypermedia as the Engine of Application State. For example, let's say we have a database with a record of employees at ACME Enterprises. By using a web portal and posting a request to add one new employee, the application changes to a state of growth by increasing the record of employees in an increment of one. Think of it in these terms: sharing representations by using self-descriptive messages to identified resources will allow you to change the state of the application.
- **Client-server:** This constraint represents itself in a simple manner which shows you that clients are separate from servers. For example, a client does not care about the storage on the server. Likewise, the server does not concern itself with the user interface or state. This allows both to be scaled, replaced, and developed independently.
  - **Stateless:** This refers to the communication between the client and server. By this definition, each request from the client to the server must contain all of the necessary data in order for the request to be understood and also so that it cannot take advantage of any stored data on the server. This means all the information is housed on the client-side. Think of CSS code in web pages, as an example of this.
  - **Cacheable:** Each response from a server to a client must explicitly define whether the data is cacheable or non-cacheable. If the response is cacheable, then the client is granted the right to reuse the response data at a later time for equivalent requests. Caching allows for increased efficiency, scalability, and user-perceived performance gains.



- **Layered system:** A layered system allows different layers to be created within an architecture so that each layer is only aware of the immediate one in which it is interacting. From the client perspective, this means that it cannot tell if it is connecting to one server on the backend, or a series of intermediary servers along the way. On the backend, layering lets legacy services to be encapsulated and protects legacy clients from receiving new services. It also aids in the overall ability to scale and load balance.
- **Code on Demand:** Code on Demand represents the only optional component of the REST architecture. It allows servers to temporarily extend or customize the functionality of the client by transferring data it can execute. An example would be JavaScript code that could prompt for user input.

## Application programming interface (API)

API is a set of instructions, which programmers can use to access web-based applications or tools. Software companies release APIs to the general public so that other programmers and developers can design products that are powered by its services.

Two examples of this would be Amazon and eBay. Since both companies release an API, developers can easily tap into Amazon or eBay's product listings, and have these products listed on a third-party website. The third-party website could embed links to these sites where purchases could be made directly, instead of visiting both sites independently.

REST APIs that adhere to the REST architectural constraints are commonly referred to as being RESTful.

The following table shows the HTTP methods that are typically used to implement a RESTful API:

Resource	GET	PUT	POST	DELETE
Collection URI, such as <a href="http://example.com/resources">http://example.com/resources</a>	List the URIs and perhaps other details of the collection's members.	Replace the entire collection with another collection.	Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.	Delete the entire collection.

Resource	GET	PUT	POST	DELETE
Element URI, such as <a href="http://example.com/resources/item17">http://example.com/resources/item17</a>	Retrieve a representation of the addressed member of the collection, expressed in an appropriate media type.	Replace the addressed member of the collection, or it doesn't exist, create it.	Not generally used. Treat the addressed member as a collection in its own right and create a new entry in it.	Delete the addressed member of the collection.

In terms of vRealize Automation, the REST API allows us to automate everything in our cloud. We can auto generate new catalog items and services, create new tenants, and customize software to fit our needs during the installation process. We can also inject commands into every stage of the provisioning process. There are numerous products and languages that allow you to leverage REST in Orchestrator. In the following sections, we limit the scope to using the built-in options that are within Orchestrator in order to create workflows.



For a detailed guide of vRealize Orchestrator, read *VMware vRealize Orchestrator Cookbook* by Daniel Langenhan through Packt Publishing

## vRealize Orchestrator configuration

VMware's workflow product is called Orchestrator. It has been packaged as part of vCenter for several years and is an important part that extends vRA's functionality. For example, we can leverage vRO to add users to Active Directory, or add additional features to our Windows 2012 R2 VMs after provisioning. Our focus will be on the vRO version that comes installed as part of the vRA appliance.

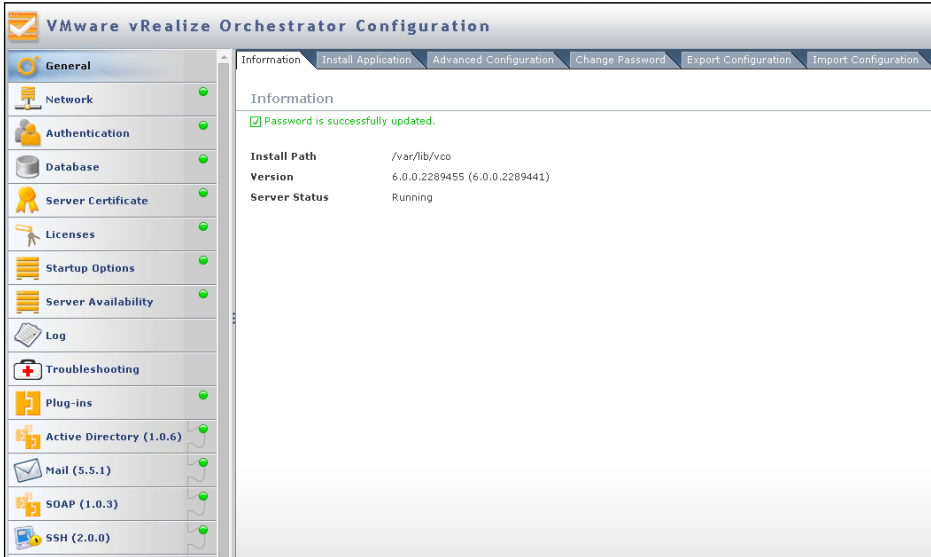
To configure vRO, log in to `https://vRA.domain.local:8283/vco-config`. The default username and password are `vmware/vmware`. Once you log in for the first time, you will be prompted to change the password. Do so, and hit the **Apply Changes** button.



If the page cannot be displayed, make sure you do not have port 8283 blocked by a firewall. Also, make sure the service is running on the vRA appliance. The easiest way to do this is the SSH into the vRA appliance, and navigate to `/etc/init.d` directory. Issue the following command:

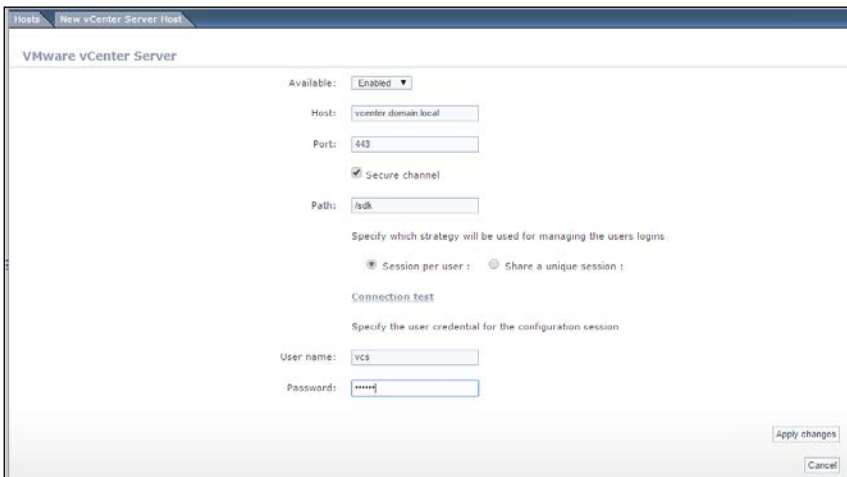
```
Service vco-configurator start
```

Once logged in, you will see a screen similar to this one:



Scroll down to the very last option on the left-hand side labeled **vCenter Server**. Verify that it has a green icon next to it. If it has a red arrow instead of a green icon, this indicates the option needs to be configured. Click on **vCenter Server** on the left-hand side and on the right, there will be an entry for your vCenter server. Click on **Edit**.

All the necessary criteria is likely to be prepopulated. You will simply need to enter the password associated with the username, and make sure **Share a Unique Session** is enabled. Hit **Apply Changes**, as shown here:



If you have no red alerts in the configurator, there is nothing you need to modify and you can exit.

## Configure the vRO endpoint in vRA web portal

In order for us to utilize the vRA APIs, we need to configure the vRO application and plugin. Follow these steps to set up the components for use:

1. Log into the vRealize Automation web portal with your `vRAadmin@domain.local` account. Click on the **Infrastructure** tab, then on **Endpoints**. Click on **Endpoints** once more on the left-hand side and you will see the vCenter endpoint we setup in *Chapter 4, Installing and Configuring vRealize Automation 6.2*, in the center of the screen.
2. On the right-hand side of the workspace, hover your mouse over the **New Endpoint** button, then select **Orchestration**, and **vCenter Orchestrator**.
3. The vRO application is built in to the vRA appliance, so it will be easy to leverage this functionality. In the **Name** field, type `vRealize Orchestrator`. You may also type a description if desired.
4. In **Address**, type in the address of the vRA appliance at port 8281 and the `vco` folder. For example, `https://vra.domain.local:8281/vco`.
5. In **Credentials**, select the same ones you used for vCenter by using the button on the right-hand side.
6. Finally, under **Custom Properties**, add the **VMware.VCenterOrchestrator.Priority** property, which has a value of 1. Hit on **OK** to save your changes, as shown here:

**Edit Endpoint - vCenter Orchestrator**  
Manage a specific endpoint.

Endpoint

\* Name:

Description:

\* Address:

\* Credentials:  ...

Custom properties: Properties (1) New Property

Name	Value	Encrypted
VMware.VCenterOrchestrator.Priority	1	No

OK Cancel

# Running the vRealize Orchestrator Client

Now, let's open a web browser and enter the address of the vRO application we entered in step 4. You will be prompted to accept the certificate, and will be redirected to this page:

**VMware vRealize™ Orchestrator™**

**Getting Started with vRealize Orchestrator**

To create and modify workflows, or to perform administrative tasks, start the Orchestrator client by using Java Web Start:

- [Start Orchestrator Client](#)

**Configure the Orchestrator Server**

To make additional configuration changes to the Orchestrator server, use the Orchestrator configuration interface:

- [Orchestrator Configuration](#)

**Develop with the Orchestrator Server**

Contains the necessary materials for developing an Orchestrator plug-in and information about using the Orchestrator SOAP and REST APIs.

- [Develop with the Orchestrator Server](#)

**More Information**

For more information about Orchestrator, see the vRealize Orchestrator documentation landing page:

- [vRealize Orchestrator Documentation](#)

**vRealize Orchestrator Resources**

- [Product Information](#)
- [Orchestrator Blog](#)
- [Community](#)
- [Support](#)
- [Plug-ins](#)

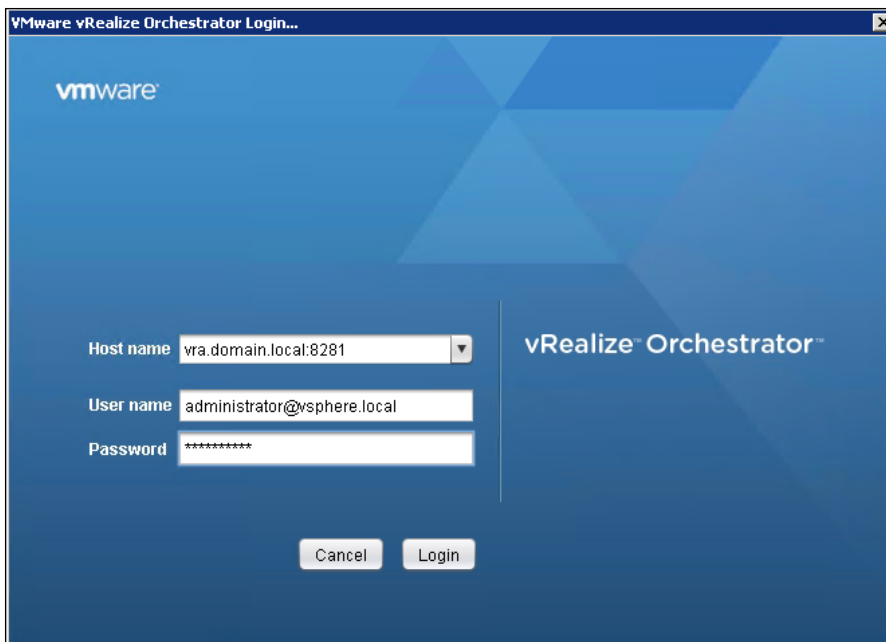
**VMware Quick Links**

- [VMware Communities](#)
- [VMware Forums](#)
- [VMware Site](#)

VMware vRealize Orchestrator 6.0.0

Copyright © 1998 - 2015 VMware, Inc. All rights reserved. This product is protected by copyright and intellectual property laws in the United States and other

Click on **Start Orchestrator Client**. This will launch a Java-based application, so make certain Java JRE is installed on the machine on which you launch the browser. You will see the following login screen. Be sure to login with your `administrator@vsphere.local` account for initial configurations:



The vRO screen is split into two workspaces, with the navigation and menu options on the left-hand side. The main workspace is on the right-hand side.

You may click on the **Permissions** tab in the right-hand workspace towards the top and add additional users. This is not required, but may be useful if you plan on having multiple vRO administrators and workflow developers.

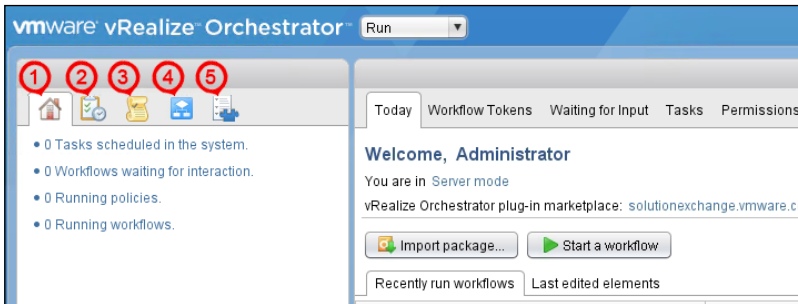
In the following few sections, we will cover navigating through the vRO client, as well as executing some API calls.



Please remember that you will see lots of name and code references called vCAC while exploring vRO. There is a lot of legacy code that will remain, and these are not typing mistakes.

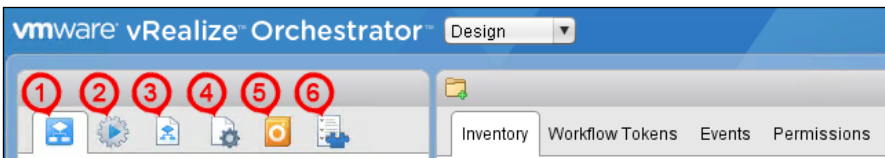
# Navigating through vRealize Orchestrator Client

Now that we are logged into the vRO client, let's familiarize ourselves with the options. By default, you are in the **Run** mode, which is one of the three available modes. The other two are **Design** and **Administer**. You can switch between them by selecting the drop-down button to the right-hand side of **vRealize Orchestrator** in the upper left hand-hand corner of the screen. Starting in the upper left-hand corner, there are five icons in the default **Run** mode, as shown here:



1. **My Orchestrator:** think of this as the Home button
2. **Scheduled Tasks:** This will be blank by default, but will list scheduled tasks once you have configured them
3. **Policies:** Use this option to create new policies to govern your vRO operations
4. **Workflows:** This is where you will do the bulk of development with the APIs
5. **Inventory:** This contains all of the out-of-the-box operations you can perform

When you switch to the **Design** mode, the following icons appear in the same location as mentioned earlier:



1. **Workflows:** This is where you will do the bulk of development with the APIs
2. **Actions:** This sub header contains preconfigured scriptlets for action items, such as inserting the date, fetching VMDK information, and connecting to LDAP servers
3. **Resources:** This contains items, such as icons and workflows from PowerShell, SQL, vCenter, and vRO, among other things.
4. **Configurations:** This contains elements for vROPs integration and vRA extensibility
5. **Packages:** This section contains preconfigured packages for utilizing items in workflows, such as PowerShell. You can import packages created by you or third-party vendors and developers to provide greater functionality to vRO
6. **Inventory:** This contains all of the out-of-the-box operations you can perform

The last mode we will explore is the **Administer** mode. Switching to this mode, the following icons appear in the same location as mentioned previously:



1. **Inventory:** As we previously explained, this contains all of the out-of-the-box operations that you can leverage in vRO
2. **Policy Templates:** These contain precreated policies for items, such as SNMP and vROPs governance
3. **Authorizations:** You can add LDAP users who are authorized to utilize vRO
4. **Packages:** This section contains preconfigured packages that are meant to utilize items in workflows, such as PowerShell. You can import packages created by you or third-party vendors and developers to provide greater functionality to vRO

Now that we have a basic understanding of how to navigate around vRO, let's create a customized workflow that leverages the REST API.



## Defining our example customized workflow

In the next section, we are going to delve deeper into vRO. Just like the other vRA topics that have been covered in this book, it should be no surprise that vRO is a complex design and execution tool. In order to make it a bit easier to digest the topics we will be covering in the section below, let's take a high level look at what we will be covering for the remainder of the chapter:

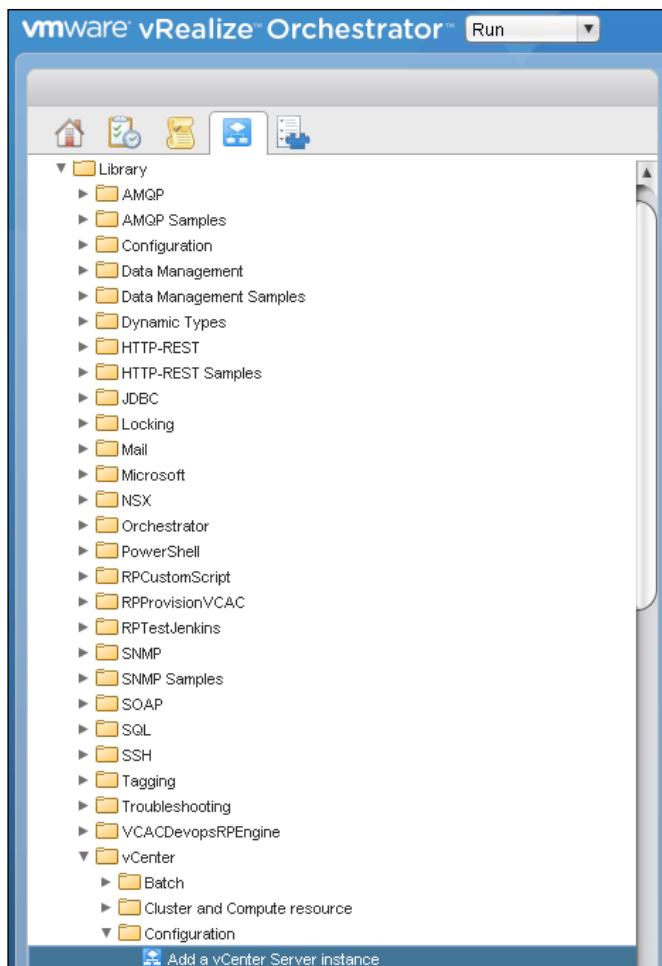
- We are going to cover how to set up the basics of designing a simple customized workflow that checks if VMware Tools is installed, then looks for a text file on the OS called complete.txt. If the text file is located, then the workflow is complete. If not, wait for a period of time and retry until the file appears.
- We will also create a workflow to enable CPU and Memory Hot Add for the VMs and present it as a service for end users to utilize in the vRA web portal.

## Configuring the prerequisites in vRO

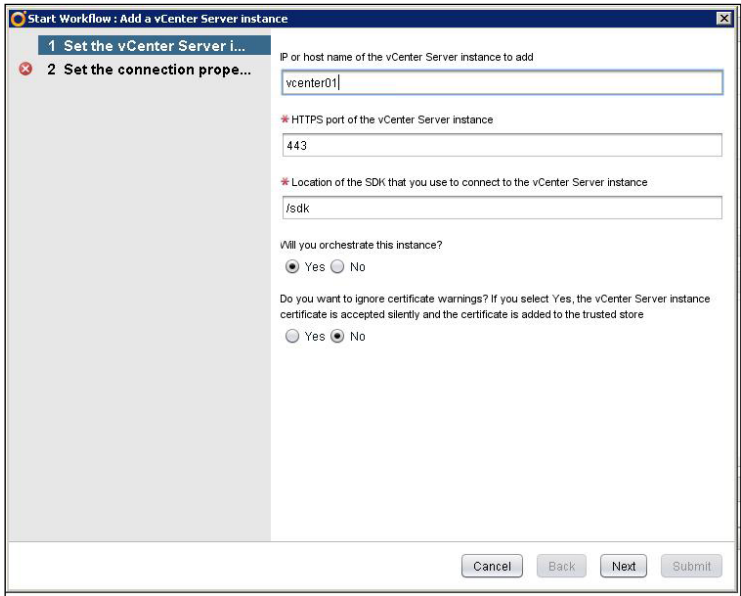
There are a few prerequisites that must be configured before we get into the creation of workflows:

- Configure vRO so it recognizes vCenter
- Add a vCAC host into vRO
- Add an IaaS host into vRO
- Install vRO customizations

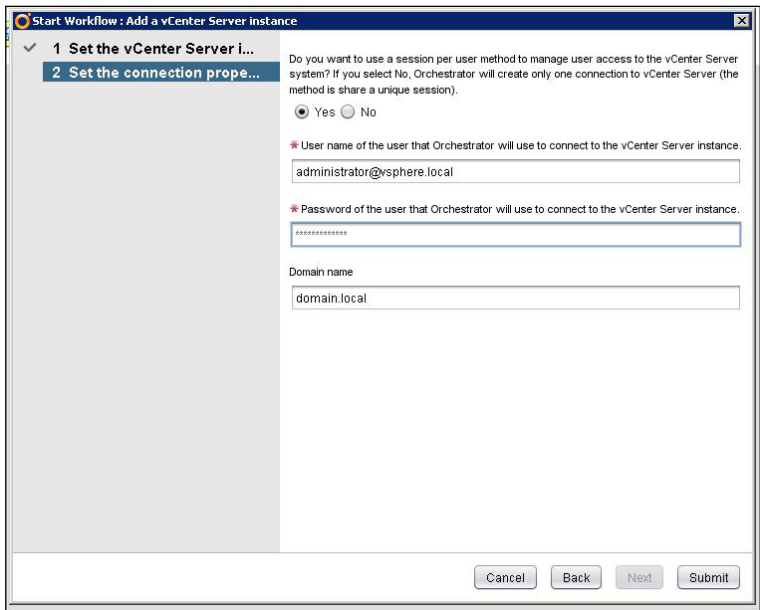
First, vRO must be aware of our vCenter installation. To configure this setting, in **Run** mode, click on the **Workflow** icon. Under your username, expand **Library**, then **vCenter**. Expand **vCenter**, then **Configuration**. Under **Configuration**, run **Add a vCenter Server Instance**, as shown here:



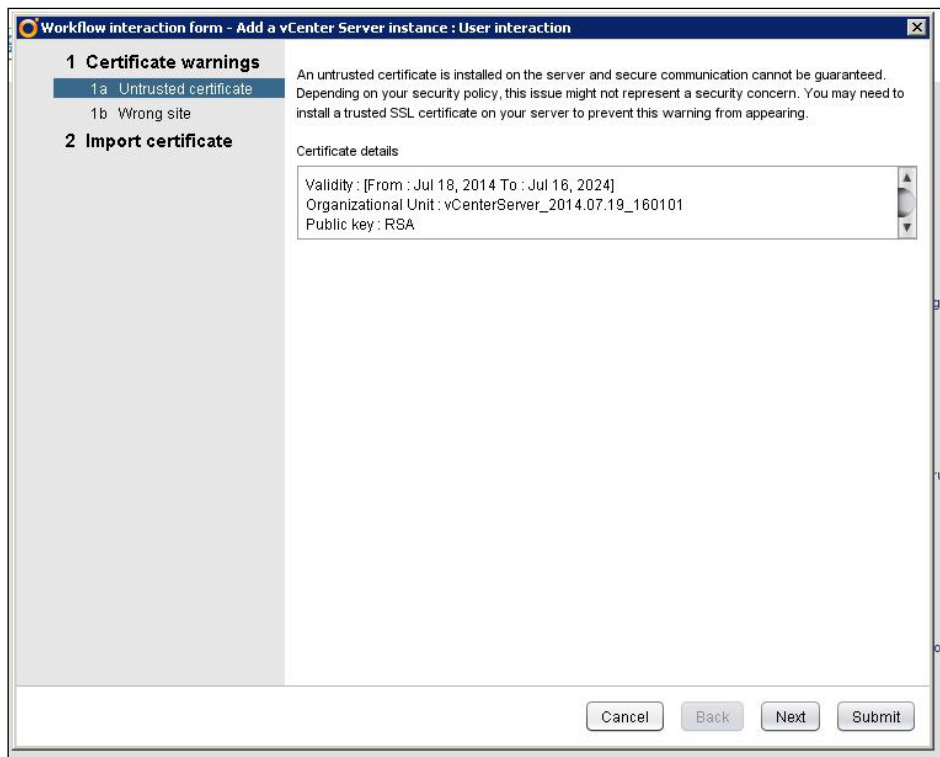
Once you run the workflow, a new dialog box will appear. Fill out the required details as listed in this screenshot, taking the default values into account:



Hit **Next** and fill in the information appropriate for your environment. If you have been following the examples, you settings will look like this screenshot:



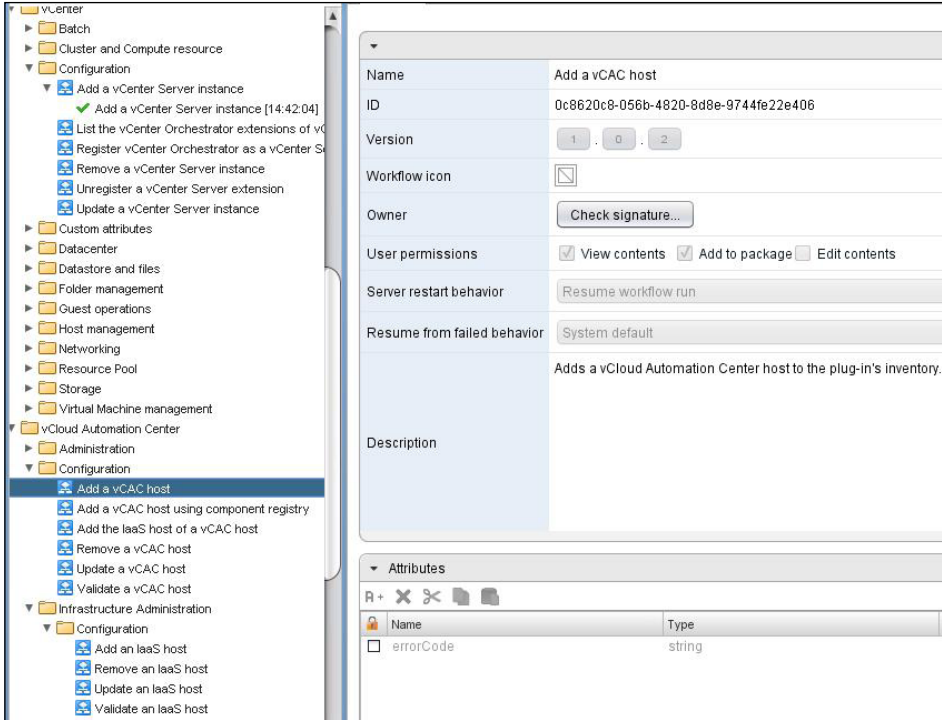
When complete, hit the **Submit** button to run the workflow. If you're using a self-signed certificate, you will get a popup like this one:



Hit **Next** to accept the default. Under **Wrong Site**, accept the default and hit **Next**. Finally, under **Import Certificate**, select **Yes**, then hit **Submit**.

When this is complete, the **bullseye** icon will appear green.

Next, we need to **Add a vCAC host**. In the **vCloud Automation Center** folder, search for **Infrastructure Administration**. Expand the folder to yield a subfolder named **Configuration**. Under **Configuration**, execute **Add the vCAC host**. Take a look at this screenshot for the location:



Click on the workflow, and then click on the green arrow that resembles a play button just above the **General** tab.

A new window will appear prompting you for input. In **Host Name**, type `vRA` and for **Host URL**, type the address of our vRA server: `https://vra.domain.local`:

Start Workflow : Add a vCAC host

1 Add a vCAC host

1a Host Properties

2 Host Authentication

2a User credentials

Properties to create a new host. The name is the host's unique identifier.

\* Host Name  
vRAC

\* Host URL  
https://vra.domain.local

Automatically install SSL certificates  
 Yes  No

Connection timeout (seconds)  
30.0

Operation timeout (seconds)  
60.0

Cancel Back Next Submit

Be sure to select **Yes** for **Automatically install SSL certificates**. Hit **Next**.

You will be prompted for **Tenant**, **Authentication username**, and **Authentication password**. Fill them out then hit **Submit**. Take a look at this screenshot for an example:

Start Workflow : Add a vCAC host

1 Add a vCAC host

1a Host Properties

2 Host Authentication

2a User credentials

If Shared Session is selected, the provided credentials are used. If Per User Session is selected, the credentials are retrieved from the currently logged-in user.

Session mode  
Shared Session

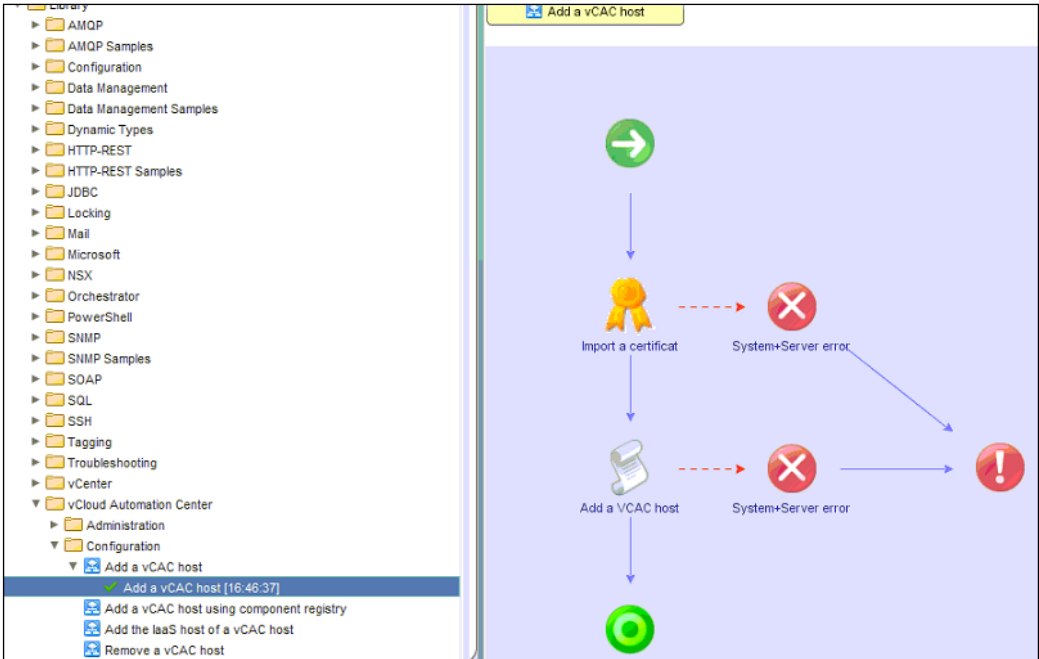
\* Tenant  
vsphere.local

\* Authentication username  
administrator@vsphere.local

\* Authentication password  
\*\*\*\*\*

Cancel Back Next Submit

Once submitted and successfully validated, the **bullseye** icon that we have seen in our examples will now turn green:



Next, we need to **Add the IaaS host**. To do so, in the **vCloud Automation Center** folder, search for **Infrastructure Administration**. Expand the folder to yield a subfolder named **Configuration**. Under **Configuration**, execute **Add the IaaS host**. Take a look at this screenshot to know its location:

The screenshot displays the vCloud Automation Center (vCAC) configuration interface. On the left, a navigation tree shows the path: **Configuration** > **Infrastructure Administration** > **Configuration** > **Add an IaaS host**. The main panel shows the configuration details for the selected task.

**Configuration Details:**

- Name:** Add an IaaS host
- ID:** 5ca252c4-b831-4746-8ffc-db3eb50f5b80
- Version:** 1 . 0 . 10
- Workflow icon:** [Icon]
- Owner:** [Check signature...]
- User permissions:**  View contents  Add to package  Edit contents
- Server restart behavior:** Resume workflow run
- Resume from failed behavior:** System default
- Description:** Adds a vCloud Automation Center host to the plug-in's inventory.

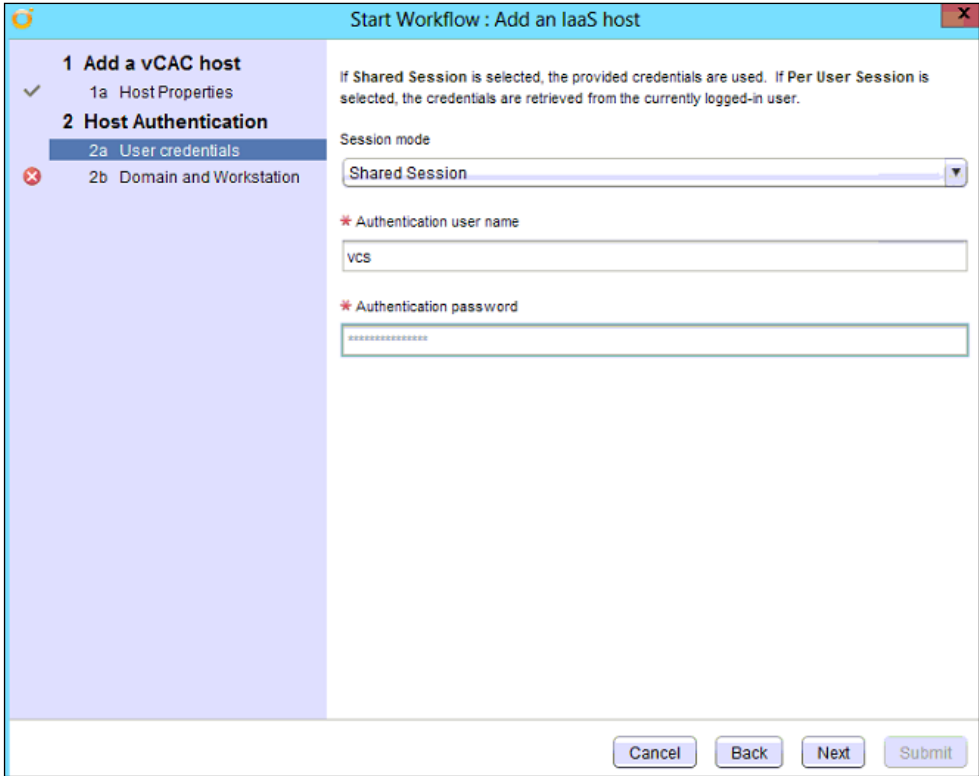
**Attributes Table:**

Name	Type	Value
<input type="checkbox"/> errorCode	string	
<input type="checkbox"/> authentication	string	NTLM


In the dialog box, you will first have to select the **vCAC** host you just added. Do this and hit **Next**. On the following screen, fill out **Name** and **Host**. Make sure you enter you IaaS host name. For our example, we use `https://vRAiaas.domain.local`. Hit **Next**.



Enter the **Authentication username** and **password**. In this section, we will use the vRA domain account, vcs. Take a look at this example and hit **Next**:



Finally, in **Domain for NTLM authentication**, enter the domain and hit **Submit**.

 Entering in the fully qualified domain name (FQDN) can cause this step to fail. If entering the domain alias causes failure, then enter `domain.local`.

As our last step in this section, we need to install **vCO Customizations**. This is going to be nearly identical to the steps we just performed. To do so, in the **vCloud Automation Center** folder, search for **Infrastructure Administration**. Expand the folder to yield a subfolder named **Extensibility**. Under **Extensibility**, execute the **Install vCO Customization**. In the dialog box, click on the **Not Set** link under the **vCloud Automation Center host** setting and select the IaaS server we had setup in the previous step. Hit **Next**.

Select **Yes** for all of the workflow stub options. Take a look at this example and then hit **Submit**:

**Start Workflow : Install vCO customization**

✓ 1 vCAC host

2 Stubs

2a State change workflow ...

3 Virtual machine menus

3a Virtual machine menus ...

State change workflow stubs to update to run vCO workflow

WFStubBuildingMachine  
 Yes  No

WFStubMachineDisposing  
 Yes  No

WFStubMachineExpired  
 Yes  No

WFStubMachineProvisioned  
 Yes  No

WFStubMachineRegistered  
 Yes  No

WFStubUnprovisionMachine  
 Yes  No

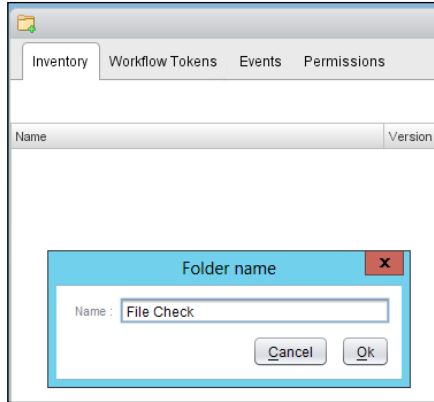
Cancel Back Next Submit

## Creating a custom workflow with vRealize orchestrator client

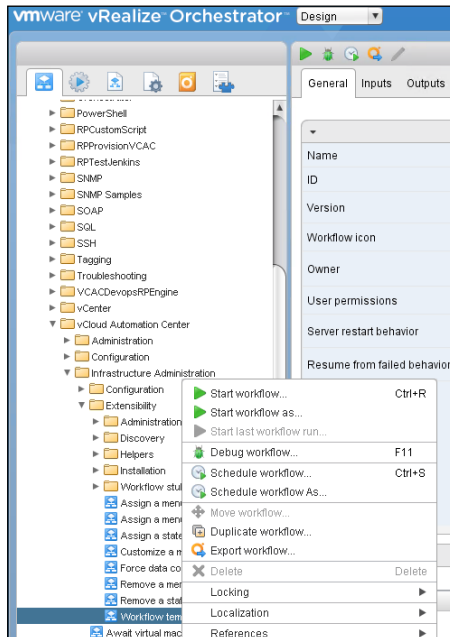
In the following example, we are going to write a workflow that allows us to check for a file within the operating system. While this seems basic, it is an excellent way to leverage VMware Tools. VMware Tools allows us to perform nearly any operation within the OS.

It is a prerequisite to have VMware Tools installed in your VM prior to executing the workflow we are getting ready to build in this section. We will not be covering how to install VMware Tools in this section, as it is assumed you already know how to complete this operation.

To create this workflow, switch to **Design** mode, and you will be in the **Workflow** section. Let's create a new folder called **File Check** for our new workflow. You can do so by clicking the **Folder** icon then typing **File Check**. Hit **OK** as shown in this screenshot:

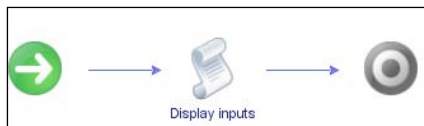


Now, we need to create a workflow. We can create one from scratch, but in this case, we are going to leverage a previously existing workflow that is out-of-the-box. Expand **vCloud Automation Center** and **Infrastructure Administration**. Finally, extend the **Extensibility** folder. You will see a workflow titled **Workflow Template**. Right-click on it and select **Duplicate Workflow** as shown here:



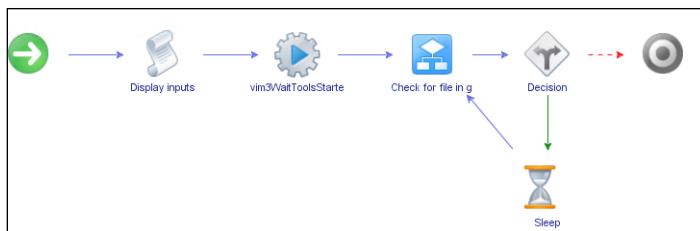
A dialog box will appear. Under **New Workflow Name**, enter `File Check`. In the **Workflow** folder, click the default option listed. A new dialog box will appear. In the upper right-hand corner is a search box. Type `File Check` and the folder we created previously will appear. Select it hit the **Select** button. You will be taken back to the previous screen; hit **Submit**.

Now, if we go back to the `File Check` folder, we will see a workflow titled `File Check`. Click on the **Schema** tab in the center of the workspace window and the following will be displayed:

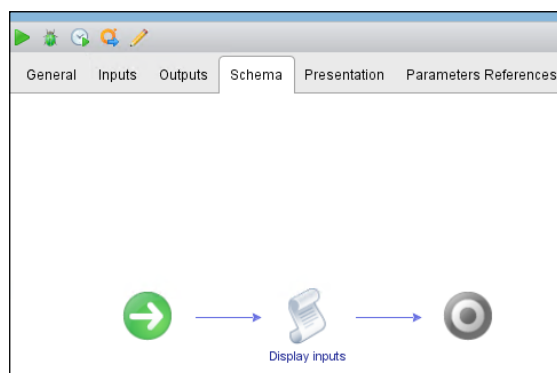


This is the basic workflow we copied. We will use this to build out the example we previously defined. The **White Arrow** in the green circle represents our starting point. **Display Inputs** will be the portion that handles logging our actions. Finally, the grey **bullseye** icon represents the final Workflow Complete state.

We are going to drag icons from the left-hand workspace into the center one. We will build upon the workflow so we have a final version similar to this one:

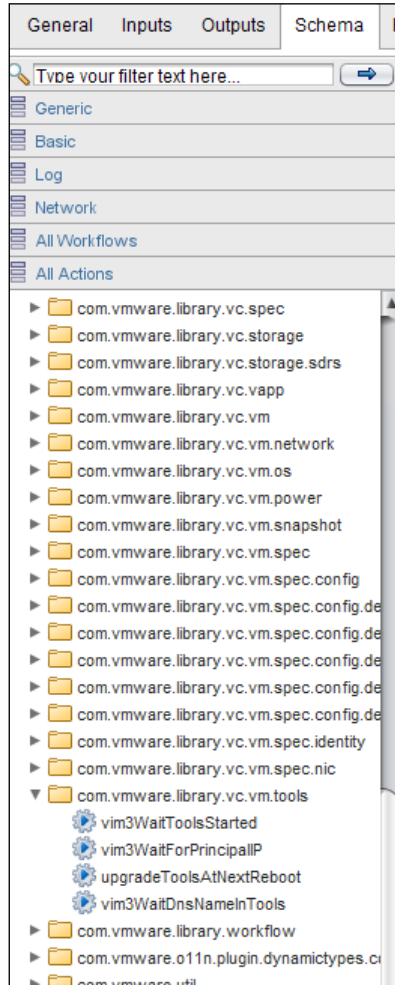


Hit the **pencil icon** to the left and above the **Schema** tab to edit the workflow, as shown here:



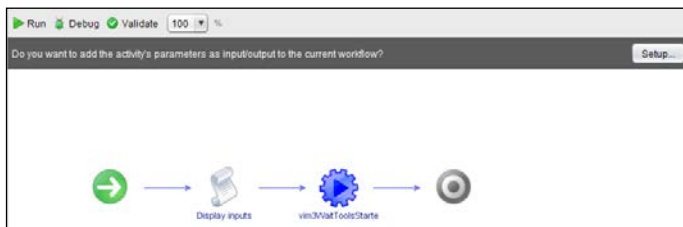
Click on **Display inputs** and the **pencil icon**. Go to the **IN** tab and delete all the options except the **vCACVm** element. Now, save your changes.

On the left-hand side, you will see several icons and categories. Click on **All Actions**, and expand **com.vmware.library.vc.vm.tools**. There will be three elements listed. Select **vi.m3WaitToolsStarted** and drag it to the right-hand side of the **Display Inputs** icon, as shown here:



Once you gain an understating of vRO and its elements, you can speed up your searches by using the search box in the upper left-hand corner. While we familiarize ourselves with the contents, we will utilize the folders and icons to find the elements we need.

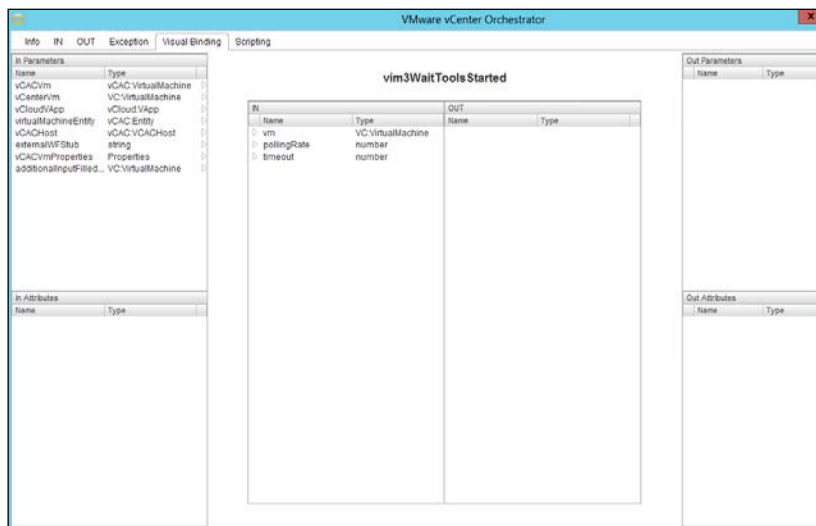
This element will wait to see if VMware Tools is installed and running within the guest VM. Once the icon is dropped into place, a prompt will appear at the top of the workspace, asking **Do you want to add the activity's parameters as input/output to the current workflow?** A **Setup** icon will appear in the upper right-hand corner, as shown here:



Instead of clicking on **Setup**, hover your mouse over the **vim3WaitToolsStarted** icon. This will reveal three additional icons, as shown here:

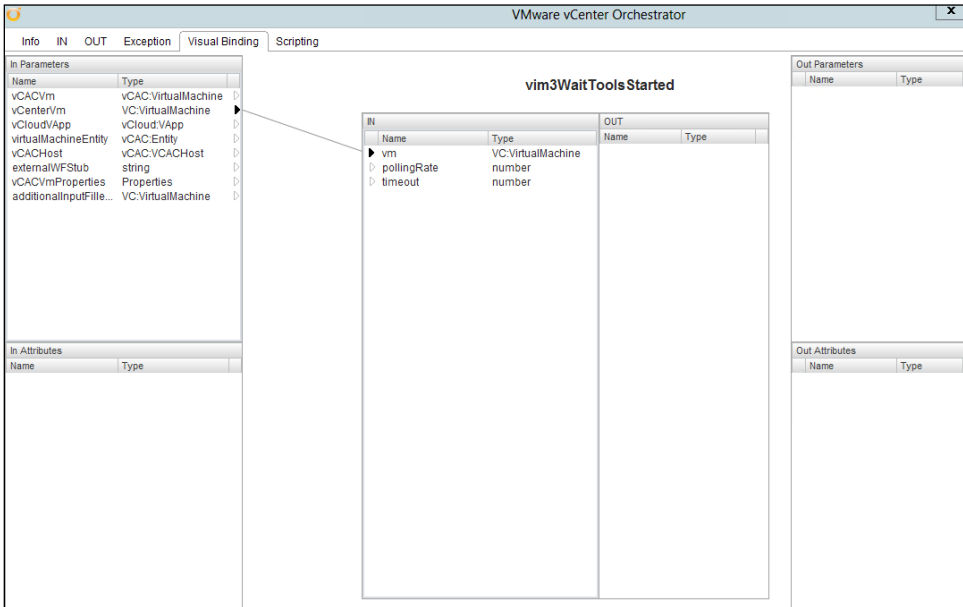



Clicking on the **pencil** icon will allow us to edit this function. Once you do this, a new dialog box will appear. Click on the **Visual Bindings** tab. Think of this as the wiring diagram for your workflow. We will need to connect the various components together, as shown here:



The dialog box is separated into five separate panels, with the one in the center representing the heart of the workflow. To explain these sections in more detail, **In Parameters** represents the options of items coming into the workflow. For example, we already know what the **vCenterVm** parameter is, as it will be defined by the Blueprint or Build Properties in the vRA web portal.

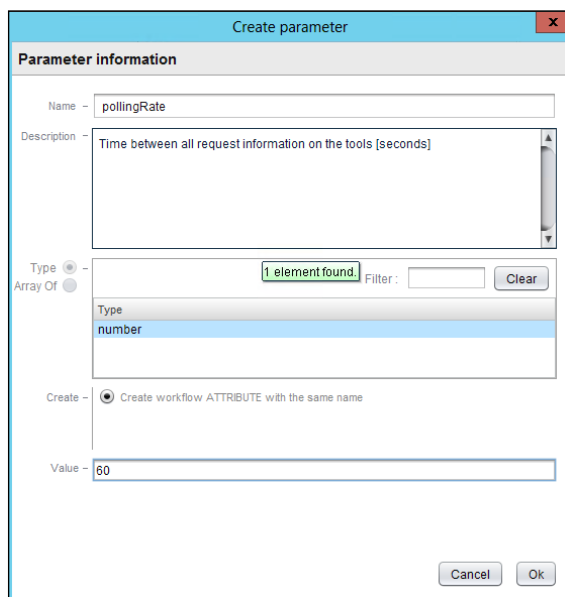
To connect this workflow, highlight the parameter on the left-hand side and drag it over to the same value in the center panel. Make sure the **Type** of each panel matches the others. In this example, the **Type** is **VC:VirtualMachine**, as shown here:



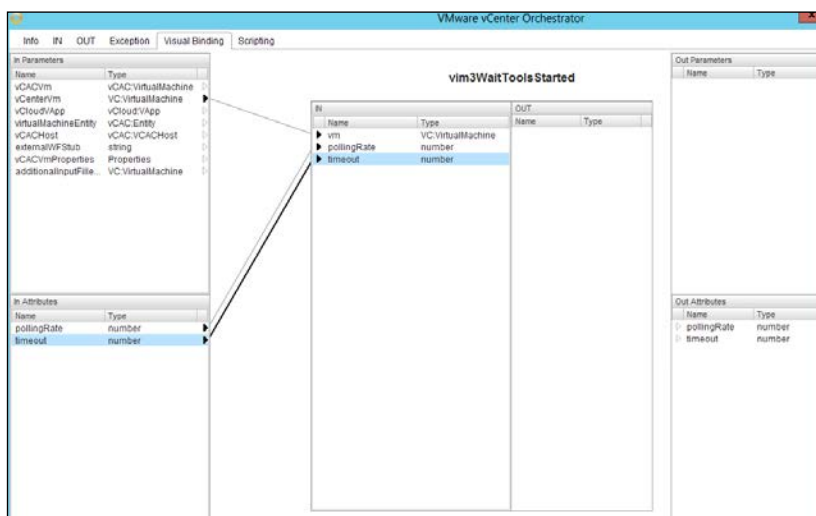
 Note that you can connect additional flows by clicking on an element in one column, then by holding the left button of your mouse, drag a line to an element in another column. You can remove the connection by clicking in the center of the line and then clicking the red X icon.

We do not know what the values for **pollingRate** and **timeout** will be, so we will need to define them. Drag the two elements one by one into the **In Attributes** panel. This panel allows us to define additional attributes for inbound options. The same is true for **Out Parameters** and **Out Attributes**.

Once you drag the element over, a dialog box will appear. Give **pollingRate** a value of 60. This is defined in seconds, so the workflow will check once per minute every minute, until it determines that VMware Tools is installed. Hit **OK**, as shown here:



Repeat the process for **timeout**, and provide a value of 60 as well. This means after one hour the workflow will stop looking for VMware Tools to be installed. Also, notice that connections will be made from the center panel to the **In Attribute** panel at the bottom, as shown here:

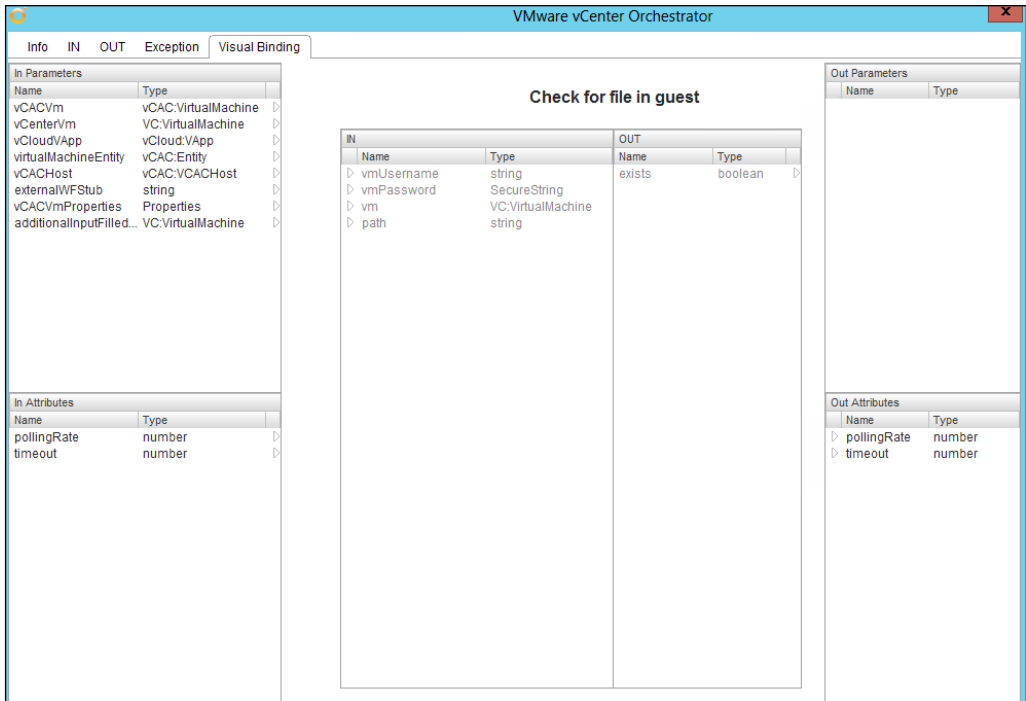




Once the tasks above are complete, hit **Close** in the bottom right-hand corner and you will return to the design screen.

Now, we need to add a step in the workflow to check for the `complete.txt` file in our VM. On the left-hand side, select **All Workflows**. Under the username you are logged in as in vRO, expand the **Library** folder. Expand **vCenter**, **Guest Operations**, and **Files**, and drag and drop the check for the file in guest and place it to the right of **vim3WaitToolsStarted**.

Click on the **pencil** icon and a new dialog box will appear. Click on the **Visual Bindings** tab and you will see the options, as displayed in this screenshot:



We will need to wire up some additional options. In this example, just as in the previous steps, we already know the VM name, so connect **VC:VirtualMachine** in the upper left-hand panel to the center.

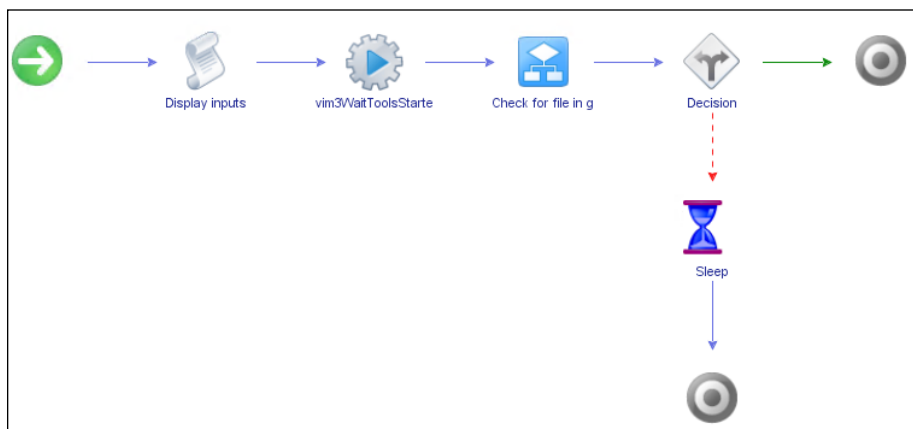


As you create more complete workflows, you will run out of space in the **Attributes** column. To prevent accidentally linking an unwanted value with an attribute, drag the attribute to the type header in the column.

Next, we need to define **In Attributes** for **vmUsername**, **vmPassword**, and **path**. The elements for **vmUsername** and **vmPassword** will be the username and password of the administrative user who is associated with the VM being provisioned. **Path** will reference where the file we will be searching for is located. In our example, we will enter `C:\Windows\Temp\complete.txt`. Enter the values that are appropriate for your environment in the **Value** section and hit **OK**.

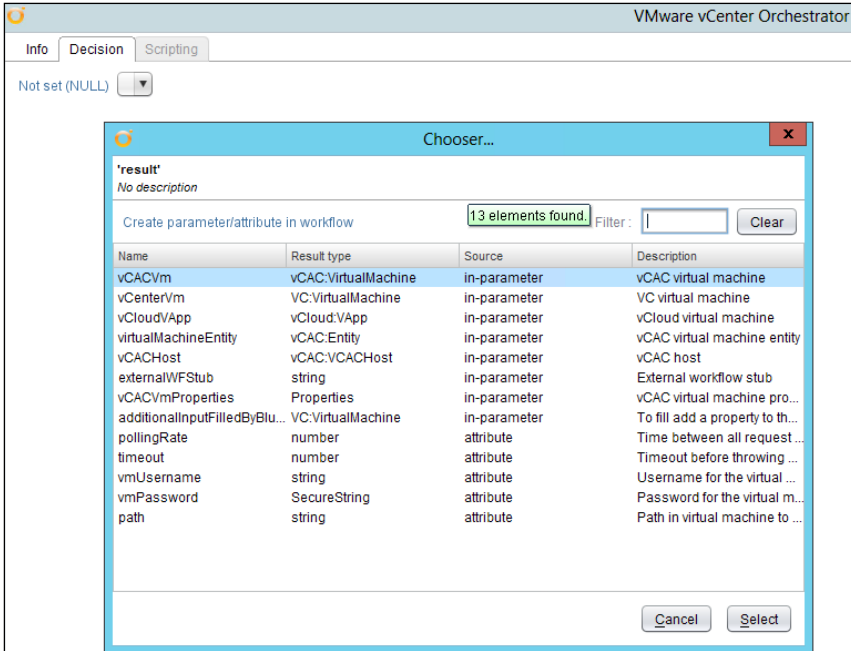
In the center panel in the **Out** column, drag **exists** to the **Out Attributes** section. Take the defaults and hit **OK**. Once this is complete, hit **Close** in the bottom right-hand corner and you will return to the design screen.

Now, we need to add an element for **Decision**. This is located under **Generic**. Drag and drop the **Decision** icon to the right of the **check for file in guest** icon. Under **Basic** on the left-hand side, drag the **Sleep** icon under the **Decision** icon. This will locate it between the **Decision** icon and the new grey bullseye icon, as shown here:



Delete the second **bullseye** icon that was created below the **Sleep** icon. Do this by right-clicking on the blue path between the two icons.

Click on the **Decision** tab. Next, click on **Not set (NULL)** and a new dialog box will appear. Select **Boolean** from the list and set it to **false**, as shown here:

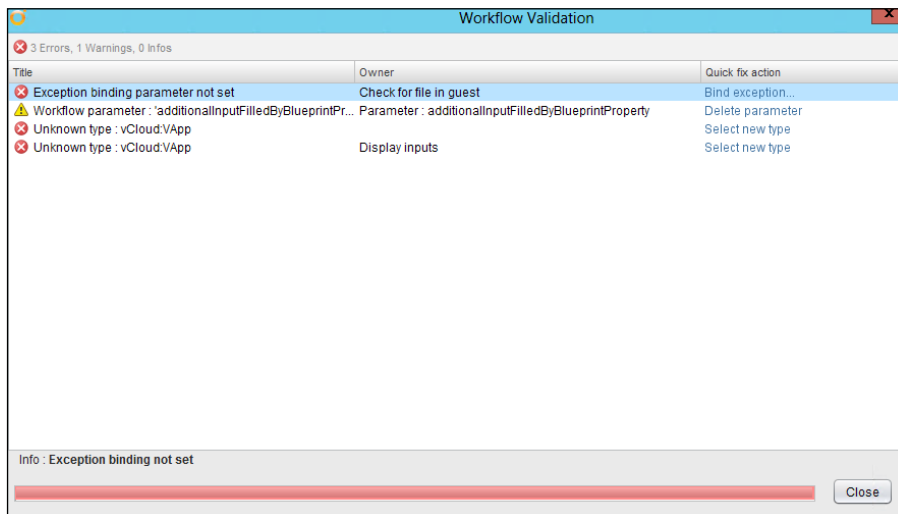


Setting the option to false means that if the file is not present, we want the workflow process to sleep for a duration of time, then try again. By intentionally creating a loop effect, we ensure the workflow will not quit running until it finds the `complete.txt` file on our VM. Once the preceding task is complete, hit **Close**.

Finally, define the duration of time the workflow should sleep for between attempts to check for the file. Edit the Sleep option and provide a **value** of **120** after dragging the **sleepTime** element to the In Attributes column. Hit **Close** when complete.

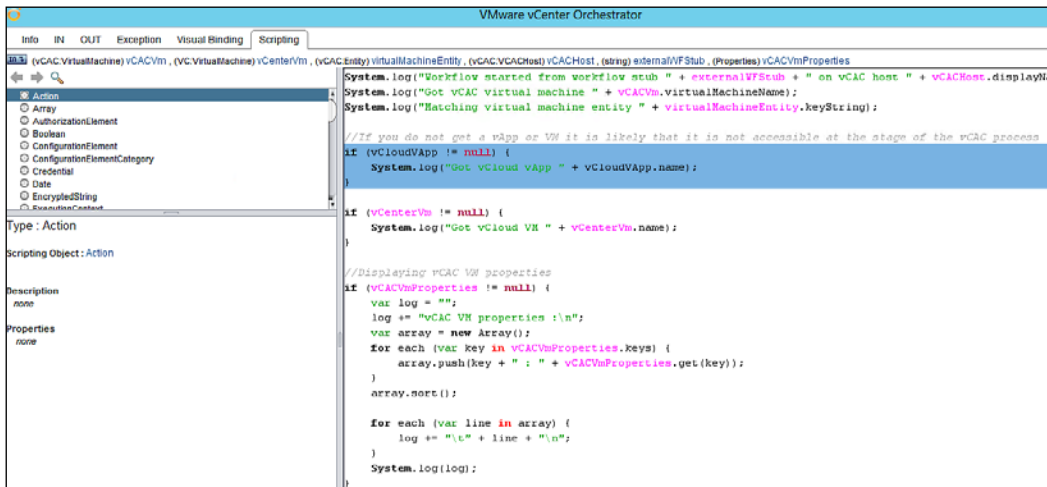
It is now time to validate our workflow. This is achieved by clicking the **Validate** button above the workflow. Since we cloned this example originally, we will get errors, but this is fine. vRO does a great job of displaying what is wrong and how to resolve it.

Take a look at this example:

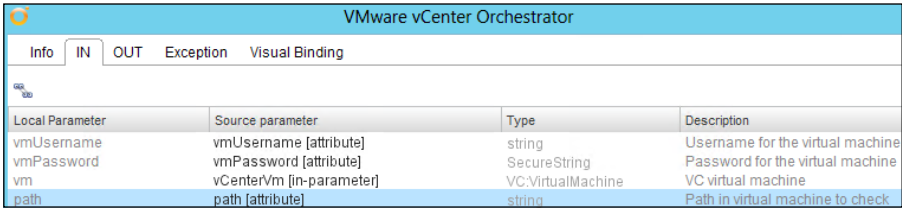


We can click on **Delete Parameter** and resolve one of the four errors. Since we are not using this parameter as part of a task or decision, it is not needed. Additionally, we are not using any vCloud vApps as part of the workflow. These options can be deleted as well, but not as easily. To do so, edit the **Display Inputs** element. Click on the **IN** tab and **vCloud vApp** will be listed. Simply delete the line and save the change.

Click on the **Scripting** tab and delete the highlighted code, referencing vCloud vApp and save your changes, as shown in this screenshot:



For the binding error, click on **Check for file in guest** icon and select the **pencil** icon to edit. On the **IN** tab, select a path from the list, then click the bind icon in the left-hand corner above **Local Parameter**, as shown in this example:



Local Parameter	Source parameter	Type	Description
vmUsername	vmUsername [attribute]	string	Username for the virtual machine
vmPassword	vmPassword [attribute]	SecureString	Password for the virtual machine
vm	vCenter/Vm [in-parameter]	VC:VirtualMachine	VC virtual machine
path	path [attribute]	string	Path in virtual machine to check

After completing these tasks, click on the **Validate** button once more. The workflow will be validated successfully. Click on **Save and Close** in the bottom right-hand corner. A dialog box will appear, prompting to add to the version history. It is a best practice to increase the version history when making changes. However, reserve this feature until you create additional workflows and add additional complexity. We will hit **Continue Anyway** and then return to the main **Design** screen.

Switch over to **Run** mode, by selecting the drop-down arrow next to **Design** in the upper left-hand corner:

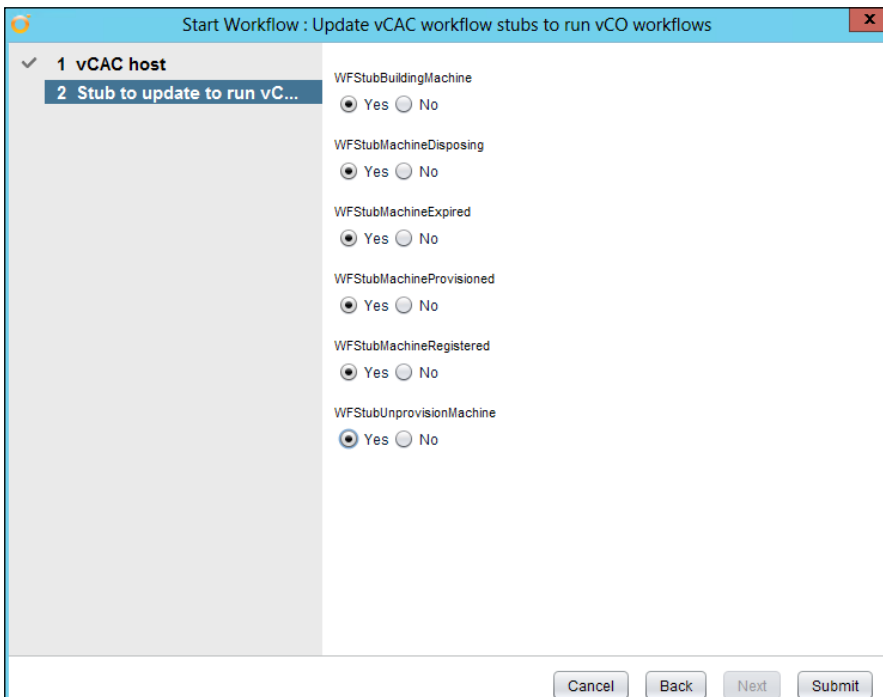


Click on the **Workflows** icon and you will see the File Check workflow we have created.

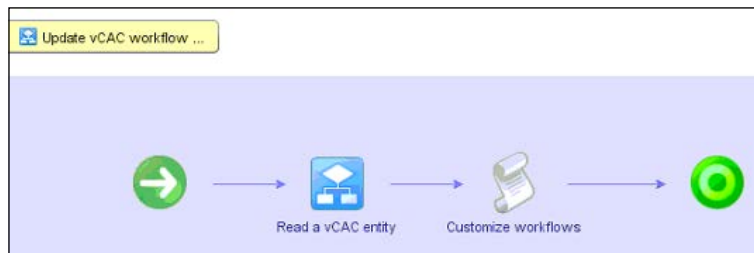
Now that our basic vRealize Orchestration configurations are complete, we will need to leverage a helper process to continue with our example. The Helper process will allow us to create the necessary steps within the vRA web portal to leverage the workflow. Look in the **vCloud Automation Center** folder on the left-hand side, then expand **Infrastructure Administration**. There is a subfolder titled **Extensibility**. Expand the folder to yield a subfolder titled **Helpers**. Expand the Helpers folder and search for a helper titled **Update vCAC workflow stubs to run vCO workflows**.

Just as we did with the **Add a vCAC host** workflow, we will need to run it and configure some options. Hit the **green arrow** that resembles a Play button to bring up the next dialog box.

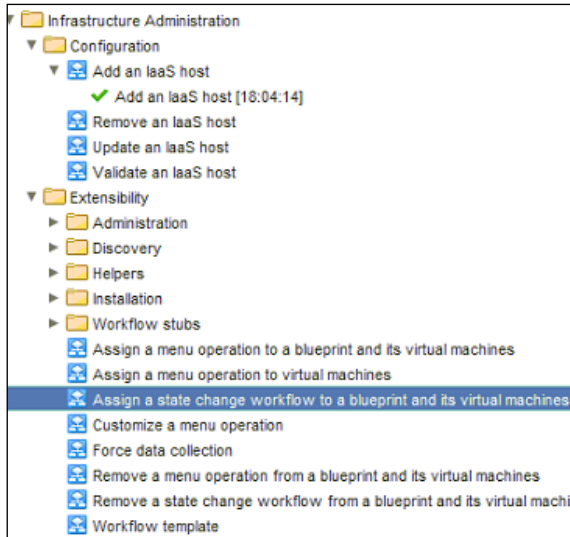
Select the vRA IaaS host we just defined earlier in this chapter, and hit **Next**. If you have two listings, either option you select will be suitable. We will change all the options to **Yes**, as this will allow us to leverage all of the workflow stubs in the future:



Hit **Submit** and validate that there is a **green bullseye** icon, as shown in the following screenshot. Note this will take up to a minute to complete:

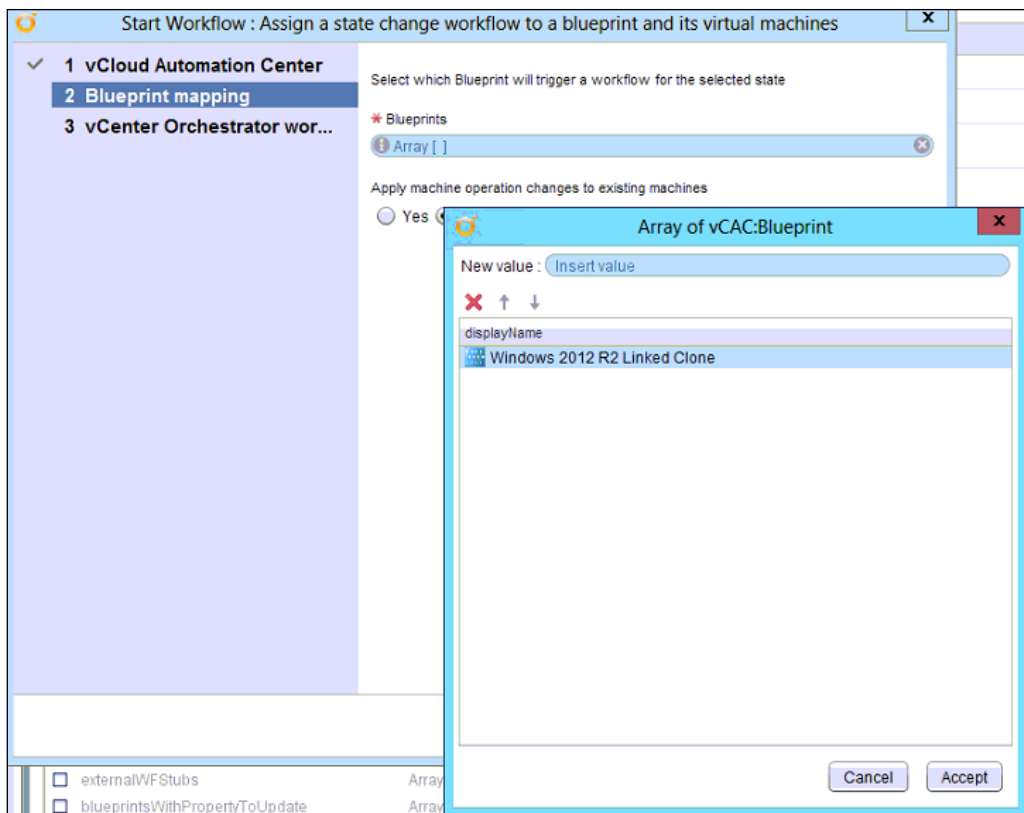


At this point, we need to configure a workflow titled **Assign a state change workflow to a blueprint and its virtual machines**. This is located in the **Extensibility** folder:



Hopefully, at this point you are getting comfortable with running and configuring the built-in workflows. Run this workflow and keep the default **BuildingMachine** workflow at the top, then select your **vCAC host** (which is really asking you for your IaaS server). Hit **Next**.

Under **Blueprints**, click on **Array [ ]**. This will open another dialog box where you will need to select your IaaS host. Expand the host entry and look for the **Blueprints** folder. Within this folder, you will see the Blueprints we configured in an earlier chapter. Select the **Windows 2012 Linked Clone** blueprint and click on the **Add** button, then hit the **Select** button. That dialog box will close and you will be back at the original Array box. Take a look at this screenshot and hit the **Accept** button:

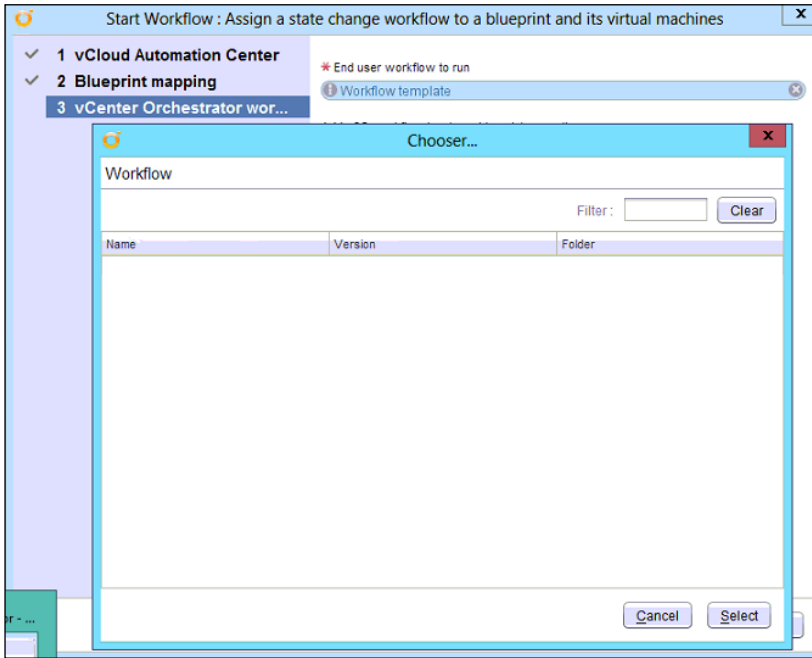


We will leave the option **Apply machine operation changes to existing machines** to **No**, as we do not want to affect any of the examples we have tested so far. Hit **Next**.

Now, we will need to enter the Workflow we created at the beginning of this section. Under **End user workflow to run**, click on **Workflow Template**. A new dialog box will appear and you can type the beginning of the name of our File Check workflow in the upper right-hand corner in the search box.



Once you select it, hit the **Select** button, as shown here:

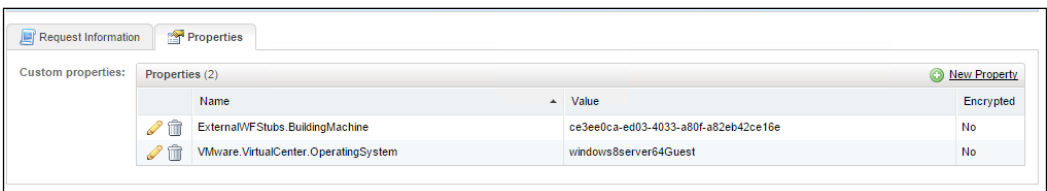


In this example, we will leave the two options under **End user workflow to run** with the default selection of **No** and hit the **Submit** button.

If you watch your screen after hitting submit, you will see the various components we configured be called under the **Schema** tab and assembled into a single custom workflow.

## Testing a customized workflow

Login to the vRA web portal and click on the **Catalog** tab. Request the Windows 2012 R2 Linked Clone item. Once requested, click on the **Properties** tab and you will see a new property based on the one created in vRO, as seen here:



This workflow will also work well with the MDT Blueprint we created in *Chapter 5, Mastering Blueprints*. Since the MDT Blueprint utilizes the `BasicWorkflow` parameter, this vRO workflow will suppress a notification from being sent to end users, stating that the VM is provisioned.



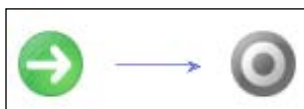
The downside to using the `BasicWorkflow` parameter, is there is no logic to inform you if the deployment was successful without adding an additional workflow through vRO, such as in the exercise above.

## Create a CPU and memory hot add workflow

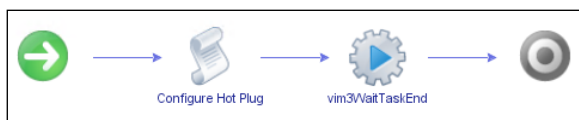
The example in the preceding section gave us the ability to check for a file in the operating system once VMware Tools has been installed. Now, let's create a workflow to enable CPU and Memory Hot Add. This will give us the ability to add a CPU and memory, without having to power down the VM.

To create this workflow, switch to the **Design** mode, and you will be in the **Workflow** section. Let's create a new folder called `CPU Memory Hot Add` for our new workflow, in this manner:

1. Click on the **Folder** icon, and type `CPU Memory Hot Add`. Hit **OK**.
2. Now, we need to create a workflow. Right-click on the folder we just created and select **New Workflow** from the floating menu. We will give the workflow the same name as the `CPU Memory Hot Add` folder. Click on **OK**.
3. This will automatically place us in edit mode under the **Schema** tab. The following will be displayed in the central workspace:



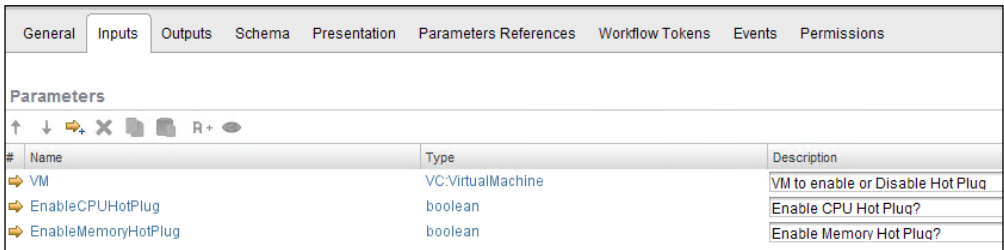
4. We are going to drag icons from the left-hand side workspace into the one in the center. We will build upon the workflow so we have a final version similar to this:



- Before dragging and dropping elements into the **Schema**, click on the **General** tab. In the **Name** Field, type CPU Memory Hot Add. At the bottom of the screen under **Attributes**, we will need to add a new attribute as it appears in the following screen. Do this by clicking the **orange R+** icon under Attributes:



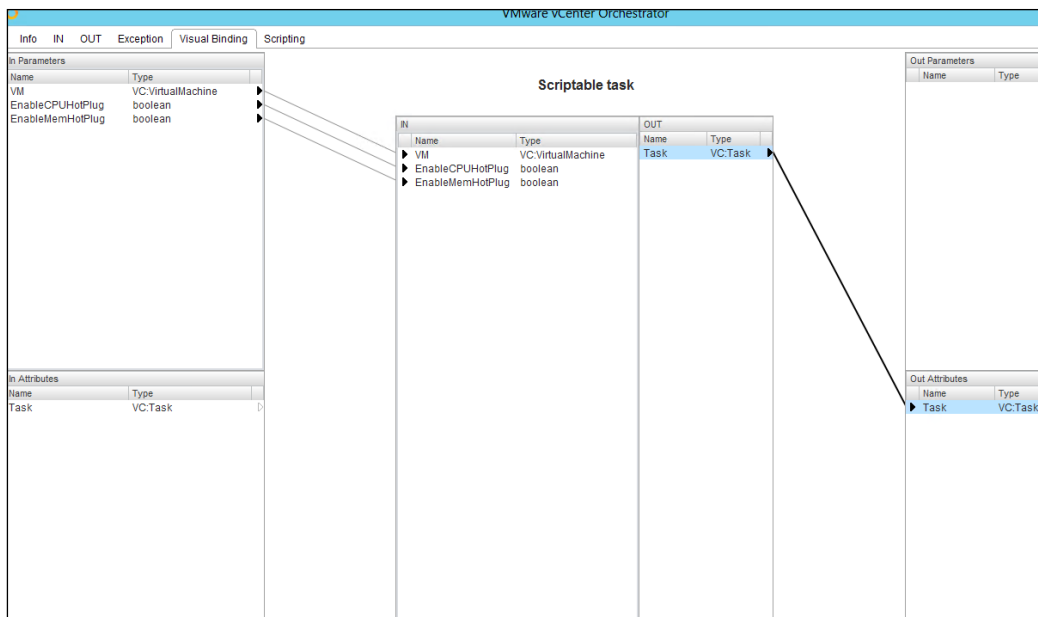
- Next, click on the **Inputs** tab. Click on the **yellow arrow** icon to add a new parameter. We will need to add three parameters total, as defined in this screenshot:



- Go back to the **Schema** Tab. On the left-hand side, you will see several icons and categories. Click on **Generic**, and select **Scriptable task**. Drag the icon between the green arrow and the grey bullseye icon.
- Hover your mouse over the **Scriptable task** icon. This will reveal three additional icons:

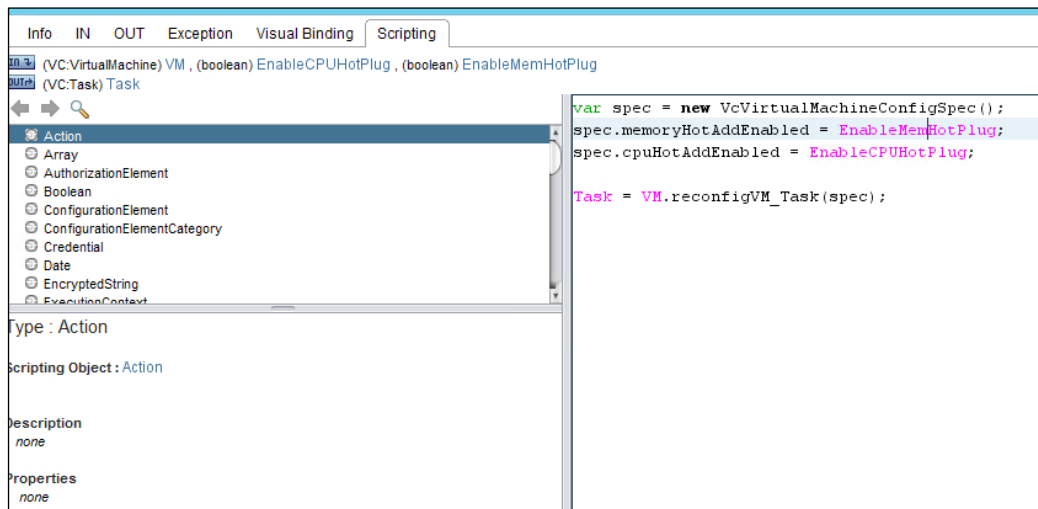


- Clicking the **pencil icon** will allow us to edit this function. Once you do so, a new dialog box will appear. Click on the **Info** tab and in the **Name** field, type **Configure Hot Plug**. Now, click on the **Visual Bindings** tab. Remember, this is our wiring diagram and in this section we will need to connect the various components together so they match this image:

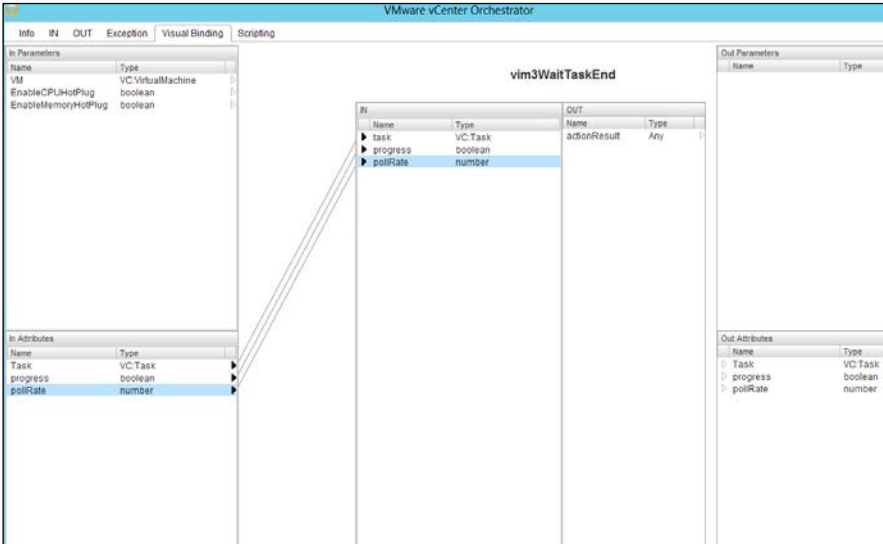


10. Next, click on the **Scripting** tab. Select **Action** on the left-hand side and paste the following in the right-hand panel so it matches the screen below:

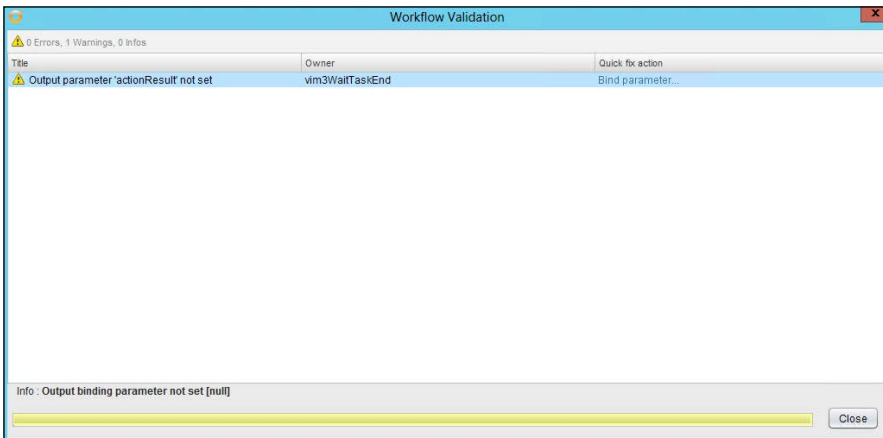
```
var spec = new VcVirtualMachineConfigSpec();
spec.memoryHotAddEnabled = EnableMemHotPlug;
spec.cpuHotAddEnabled = EnableCPUHotPlug;
Task = VM.reconfigVM_Task(spec);
```



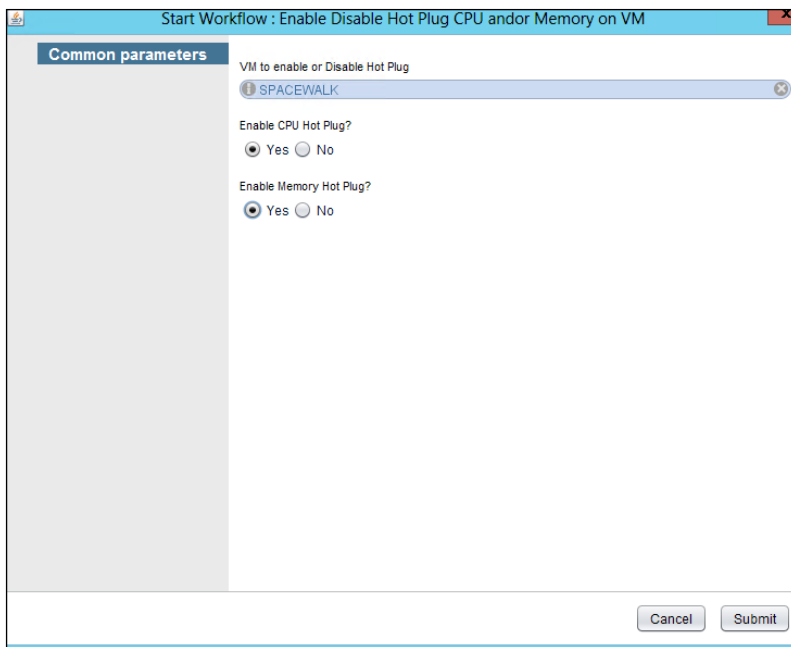
11. Hit **Close** in the bottom right-hand corner. We will be returned to the **Schema** workspace. In the left-hand corner, type the following into the search box: **vim3WaitTaskEnd**. Drag and drop this icon to the right-hand side of the **Scriptable task** we completed.
12. Click on the **pencil icon** and once the dialog box appears, click on the **Visual Binding** tab. Drag **task**, **progress**, and **pollRate** from the **IN** column in the center workspace to the **In Attributes** column. Give **pollRate** a **value** of **30**. Take a look at the screenshot:



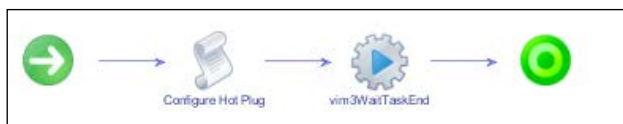
13. Hit **Close**. Once complete, hit the **Validate** button. This will bring up the following screen and error:



- Click on **Bind Parameter** and select **Null**. Next, click on the **Debug** button to make sure our workflow performs as expected. Look in your environment for a VM that does not have CPU or Memory Hot Add enabled. Choose the VM and hit **Submit**:

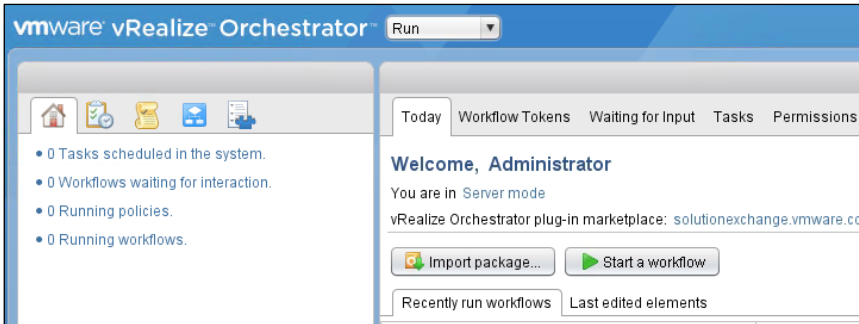


- If the workflow is successful, you will get a **green bullseye icon**, as shown here:

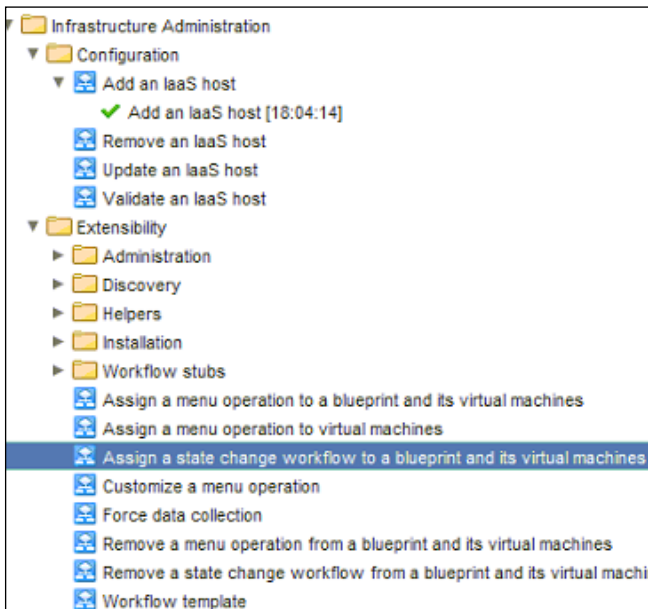


- Be sure to check in vSphere and make sure that the CPU and Memory Hot Add function is enabled in the VM you selected above as expected.
- Click on **Save and Close** in the bottom right-hand corner. A dialog box will appear, prompting you to add to the version history. It is a best practice to increase the version history when making changes. However, reserve this feature until you create additional workflows and add additional complexity. We will hit **Continue Anyway** and then be returned to the main **Design** screen.

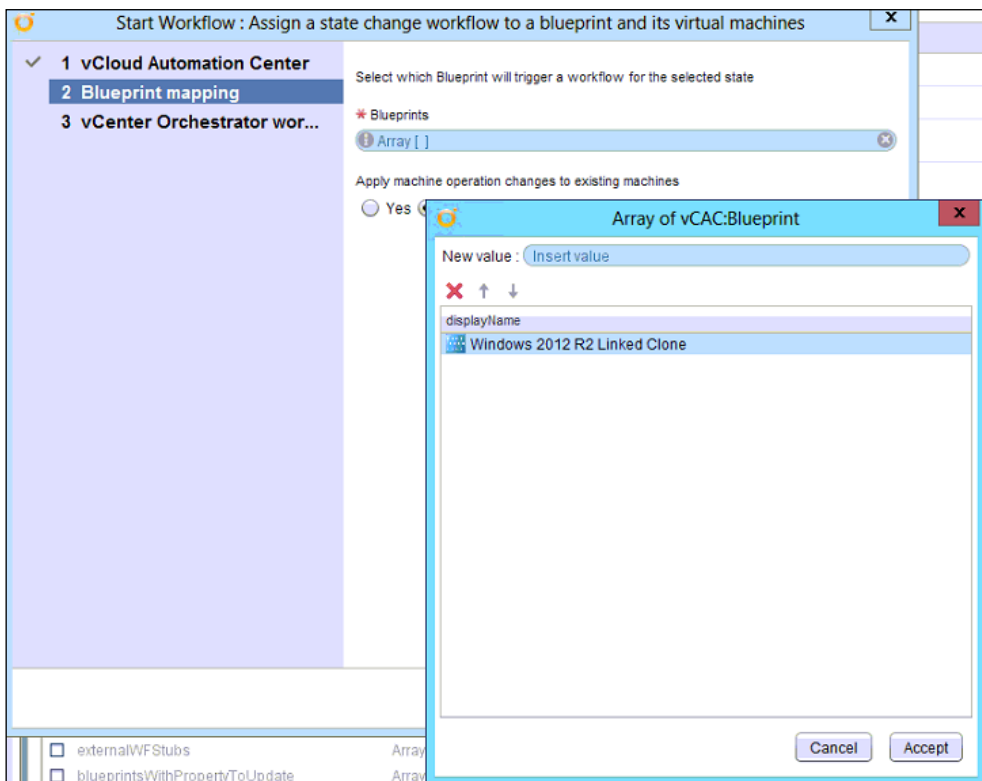
- Switch over to the **Run** Mode by selecting the drop-down arrow next to **Design** in the upper left-hand corner:



- Click on the **Workflows** icon and you will see the **CPU Memory Hot Add** workflow we have created.
- At this point, we need to configure a workflow titled **Assign a state change workflow to a blueprint and its virtual machines**. This is located in the **Extensibility** folder:



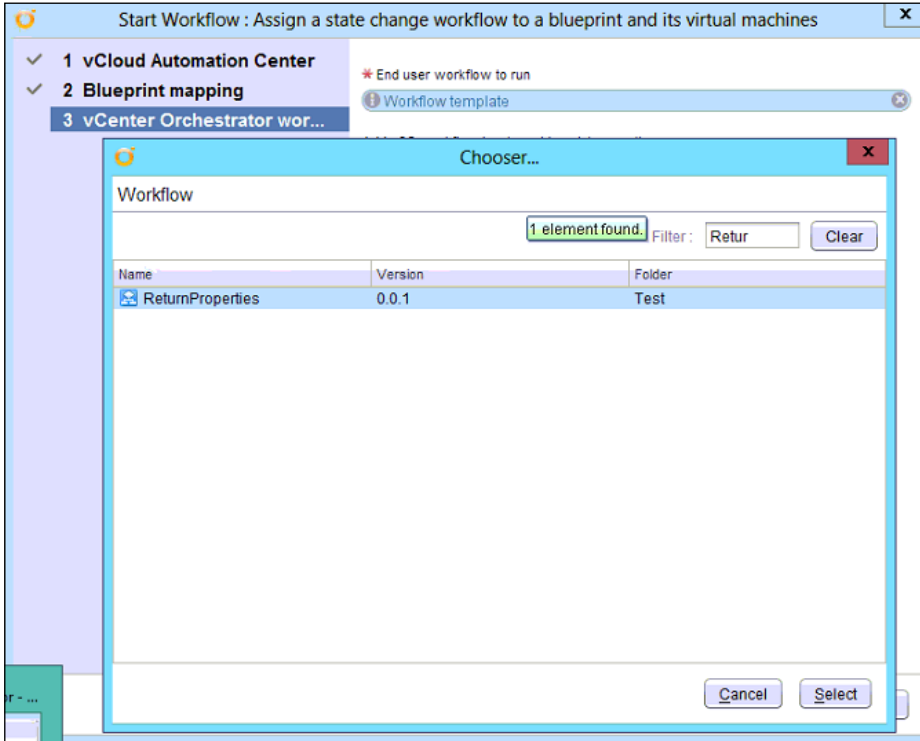
21. Run this workflow and keep the default **MachineProvisioned** workflow at the top, then select your **vCAC host** (which is actually asking you for your IaaS server). Hit **Next**.
22. Under **Blueprints**, click on **Array [ ]**. This will open another dialog box where you will need to select your IaaS host. Expand the host entry and look for the **Blueprints folder**. Within this folder you will see the Blueprints we configured in an earlier chapter. Select the **Windows 2012 R2 Linked Clone** blueprint and click on the **Add** button, then hit the **Select** button. This dialog box will close and you will be back in the original Array box. Hit the **Accept** button and take a look at the screenshot below:



23. We will leave the option **Apply machine operation changes to existing machines** to **No**, as we do not want to affect any of the examples we have tested so far. Hit **Next**.



24. Now, we will need to enter in the Workflow we created at the beginning of this section. Under **End user workflow to run**, click on **Workflow Template**. A new dialog box will appear and you can type the beginning of the name of our CPU Memory Hot Add workflow in the upper right-hand corner in the search box. Once you select it, hit the **Select** button:

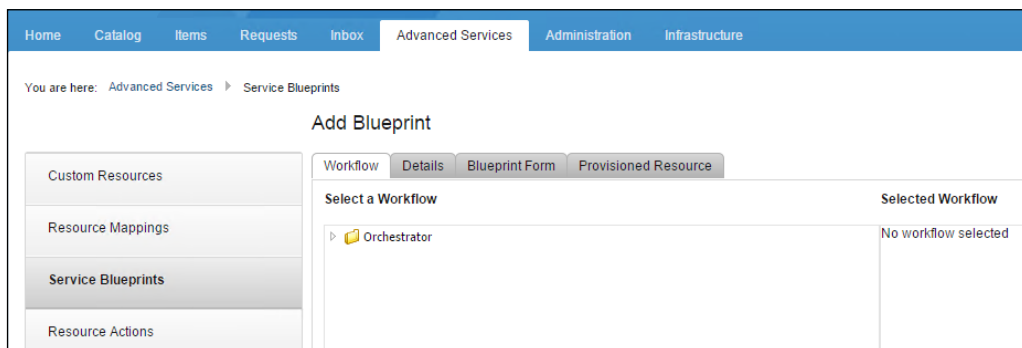


25. In this example, we will leave the two options under **End user workflow to run** with their default selection of **No** and hit the **Submit** button.

If you watch your screen after hitting submit, you will see the various components we configured be called under the **Schema** tab and assembled into a single custom workflow.

## Adding the workflows to vRA web portal

Login to the vRA web portal and click on the **Advanced Services** tab. Click on **Service Blueprints** and then on the **Add** button. A window similar to this one will be displayed:



Expand the **Orchestrator** folder under the **Workflow** tab until you get to the **CPU Memory Hot Add** folder. Click on **CPU Memory Hot Add** and then click the **Next** button. In the **Details** tab, keep the defaults and hit **Next**. In the **Blueprint Form**, we will keep the defaults as well. However, you could edit the form to allow for more or less user interaction, depending on your preference. Hit **Next**. Finally, take the defaults as well on **Provisioned Resource** and hit **Add**.

We will return to the Service Blueprint screen. Like all other Blueprints, this will need to be published as well. You can do this by hitting the **Publish** button.

In order to get our customized vRO workflow to appear on the portal for users to access, we need to go to the **Administration** tab. Click on **Catalog Management**, and then on **Catalog Items**. You will see **CPU Memory Hot Add**. Click on this item and make sure **Status** is Active and assign the proper **Service** category. When finished, click on the **Update** button.


Now, we will entitle the vRO workflow. Click on **Entitlements** on the left-hand side. Click on the Software Developers group we created earlier. Under the **Items and Entitlements** tab, click on the **plus button** to the right of **Entitled Catalog Items**. Add **CPU Memory Hot Add**. Click on the **Update** button at the bottom to save your changes.

Finally, click on the **Catalog** tab where all our deployed and entitled blueprints are listed. You will now see the CPU Memory Hot Add item we created.

## Additional workflow tools

vRealize Orchestrator comes with a wealth of great out-of-the-box tools. Often, you can reverse engineer those workflows to fit the unique needs of your environment. However, there are just as many times where this is not possible. VMware Labs developed a tool called **Onyx**, that allows you to record a manual process, such as deleting a device, and import this process into a workflow. While we will not be covering Onyx in this book, it does warrant mentioning, as it will help make custom workflow creation easier.

<https://communities.vmware.com/community/vmtn/automationtools/onyx>



You will want to define the following at the top of the Onyx code:

```
var managedObject = VM;
```

This object is not copied by default and the Onyx script will not work without it. Additionally, you will want to remove the Onyx line regarding the spec.changeVersion. This will also cause the script to fail.

## Summary

We have spent several chapters concentrating on setting up Infrastructure as a Service. In this chapter, we explored another layer making up our vRealize Automation 6.2 universe, by introducing the REST API and showing you how to leverage workflows in vRO. vRealize Orchestrator allows us to use out-of-the-box workflows and create our own. We discussed how to install, configure, and navigate around the vRealize Orchestrator client. We also configured a basic and advanced workflow, then linked it to one of our existing blueprints for testing.

In *Chapter 9, Integrating vRealize Operations Manager* we will focus on integrating vRealize Operations Manager 6 with vRealize Automation 6.2.

# 9

## Integrating vRealize Operations Manager

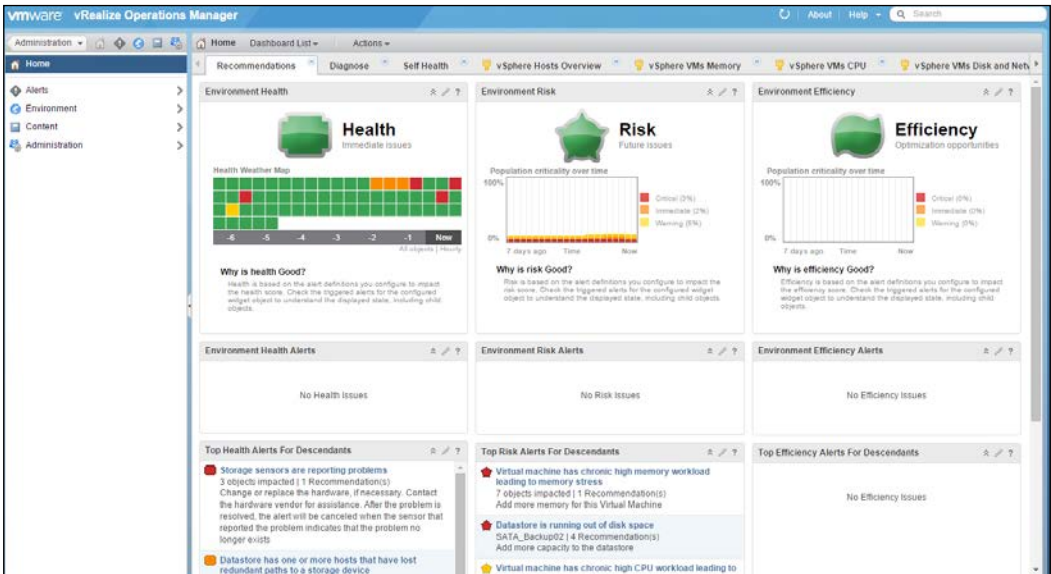
In the previous chapters, we explored everything about vRealize Automation from its history, to its design, features, and step-by-step instructions provided to install its components. With that being stated, it is fitting to begin wrapping up our adventure by discussing one of the most popular integration features. Though new to vRealize Automation 6.2, **vRealize Operations Manager 6.0 (vROps)** not only integrates with vRealize Automation, but allows us to decommission provisioned servers based on idle time.

We will be covering the following topics in the chapter:

- Overview of vROps 6.0
- Benefits of integrating vROps with vRealize Automation
- How to integrate vROps with vRealize Automation
- Utilizing vROps integration to reclaim idle virtual machines

# Overview of vRealize Operations Manager 6.0

vRealize Operations Manager is the newly branded name of vCenter Operations Manager (vCOPs). vCenter administrators are probably familiar with the product as it provides a wealth of knowledge about monitoring, trending, analysis, and alerting for the entire vCenter universe. It is distributed in the form of an OVF template and configured via its own web instance. Once you login, vROps 6.0 looks similar to this screenshot:



Though we will not go into the details of installing, configuring, or maintaining vROps in this book, you can get more information by viewing VMware's official installation guide available at <http://pubs.vmware.com/vrealizeoperationsmanager-6/topic/com.vmware.ICbase/PDF/vrealize-operations-manager-60-vapp-deploy-guide.pdf>. You can also refer to *Mastering vRealize Operations Manager*, Packt Publishing by Scott Norris and Christopher Slater.

---

## Benefits of integrating vROps with vRealize Automation

One of the major benefits of integrating the two products is to provide health insight to end users through a single product. Normally, these views are reserved for administrators and have to be viewed through the vCenter web client or the vROps portal. Once integration is complete, end users get to see the same health badge through the vRealize Automation portal.

You may wonder why this is advantageous for an end user. If you recall, we referenced the software developer group for the fictional company ACME Enterprises in *Chapter 3, Designing and Building a vRealize Automation 6.2 Infrastructure*. We want the members of the software developer group to create the servers they need in order to continue to drive development, without needing to wait for the IT department.

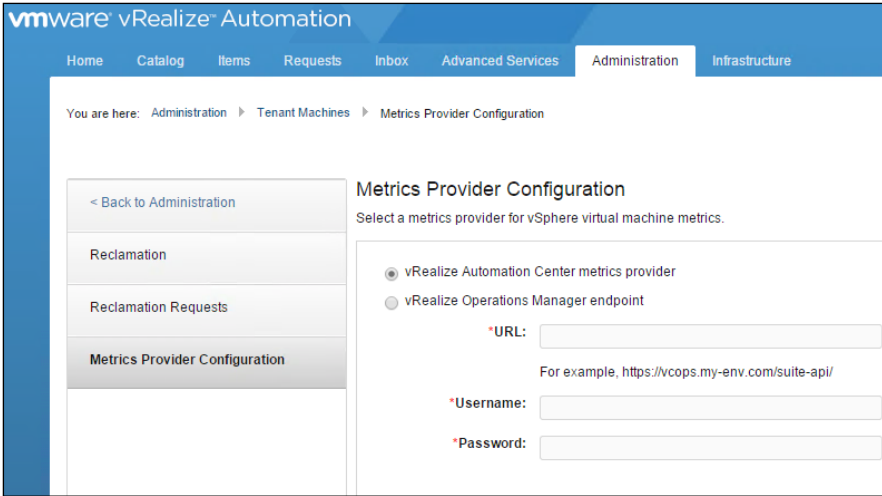
We allow them to spin up infrastructure as needed through the vRA web portal, and essentially control these components within the boundaries we define as vRA administrators. However, they do not have a way to view how their components of the infrastructure are performing in vRA without integrating vROps.

Once integrated, they can view health badges through the vRA web portal. The health metrics provided can empower them to make decisions, such as provisioning more virtual machines or requesting more compute for those that may be underperforming, based on the software development load. It is quite a powerful bit of knowledge to provide end users and will hopefully steer intelligent decisions regarding their vRA consumption, because they now have the data to do so.

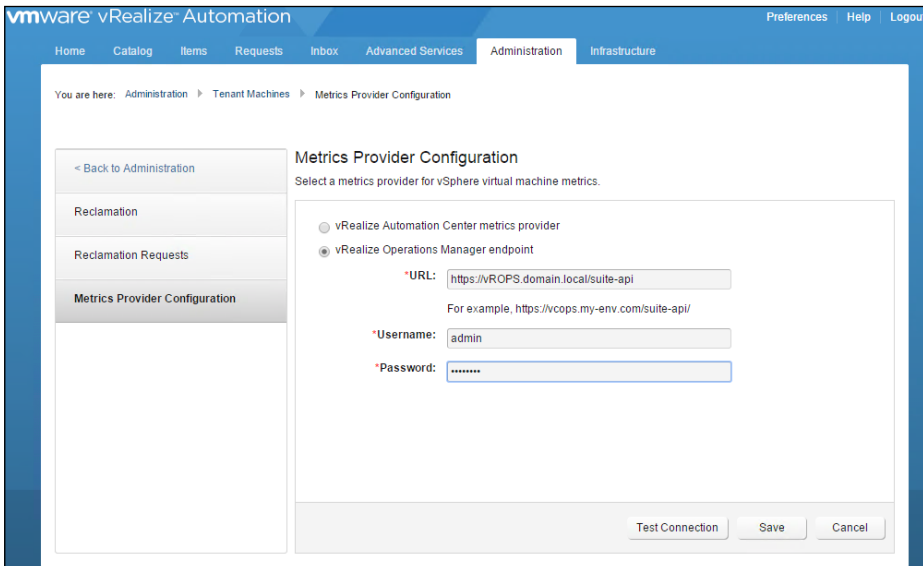
## How to integrate vROps with vRealize Automation

Now that we have an understanding of some of the benefits vROps provides, let's leverage its power by integrating it with vRA. First, login to the vRA web portal as an administrator. If you have been following the examples in the book so far, the vRA portal is located at `https://vRA.domain.local/vcac`. Browse to the **Administration** tab, then click the **Tenant Machines** tab.

Once complete, click on the **Metrics Provider Configuration** on the left-hand side of the web portal, as shown in the following screenshot:



For the examples in we this chapter, we will refer to `https://vROPS.domain.local` for the vROps installation. Select the second option from the list, which is the **vRealize Operations Manager endpoint**. Use the URL above for vROps, and append the location of the API, which is `/suite-api/`. Also, use the administrative username and password you selected when installing vROps. For our example, we will use `admin` and `P@ssword123`. See the example below:



Be sure to hit the **Test Connection** button in the bottom right-hand corner to validate your parameters are accurate. Once your connection is successful, hit the **Save** button.

Testing the integration is just as easy. Navigate to the **Items** tab, and click on a machine that you provisioned. In the bottom right-hand corner, you will see a badge from vROps. If you have used vCOPs in the past, you will be familiar with the health badges. Take a look at the following screenshot:

The screenshot displays the VMware vRealize Automation console. The main content area shows the 'Item Details' for a virtual machine named 'Demo-Win10-01'. The status is 'On'. The configuration includes 2 CPUs, 2048 MB of memory, and 50 GB of storage. The description is 'Provisioned by VMware vCAC'. The owner is 'test', and the business group is 'test'. The machine is a Windows virtual machine provisioned by VMware vCAC. The left sidebar lists various details such as 'Provisioned' date (2/14/15 2:45 PM), 'Type' (Virtual Machine), 'Cost' (\$0.00 / day), 'Lease' (Indefinite), 'Lease Cost' (NA), 'Expires' (Never), 'Cost to Date' (\$0.00), and 'Business Group' (test). The right sidebar shows a list of actions including 'Change Lease', 'Connect by Using RDP', 'Destroy', 'Expire', 'Install Tools', 'Power Cycle', 'Power Off', 'Reboot', 'Reconfigure', 'Reprovision', 'Shutdown', and 'Suspend'. Below the actions is a 'Current Health' indicator showing 'Good' with the text 'The health of this item is normal. No attention required.'

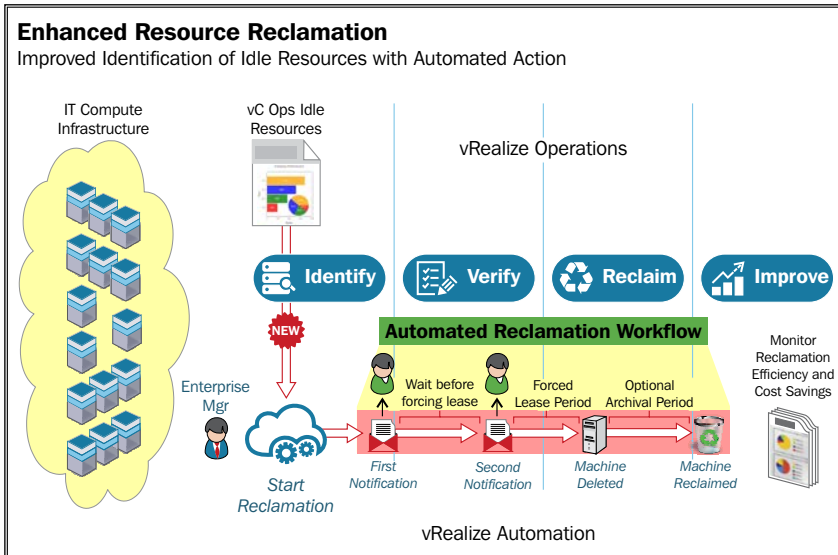
You can also view the health of the VMs in your vRealize Automation environment by navigating back to the **Administration** tab, clicking on **Tenant Machines**, and then on **Reclamation**. This is where you can review the **Reclamation metrics**, as provided by vROps.



# Utilizing vROps to reclaim idle virtual machines

The ability to remove idle virtual machines that are identified by vROps is a second thrilling addition to the vRealize Automation 6.2 release.

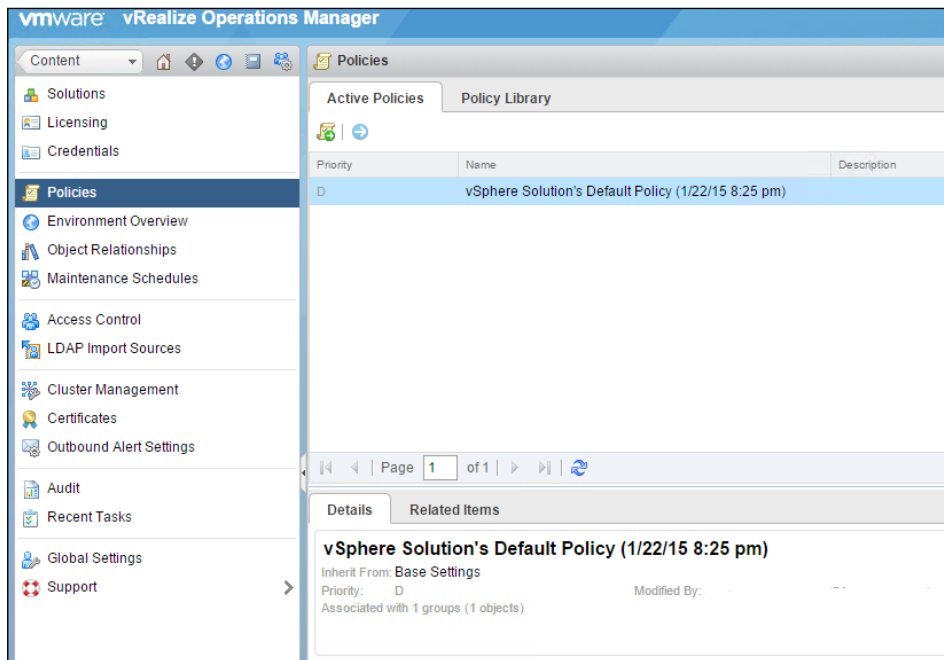
Here is an illustrated workflow of how this process works:



Once we reclaim idle VMs and compute resources, we can show direct savings back to the business units that may have provisioned the infrastructure. For example, if we assign costs to our compute resources, we can present a report showing the amount of savings.

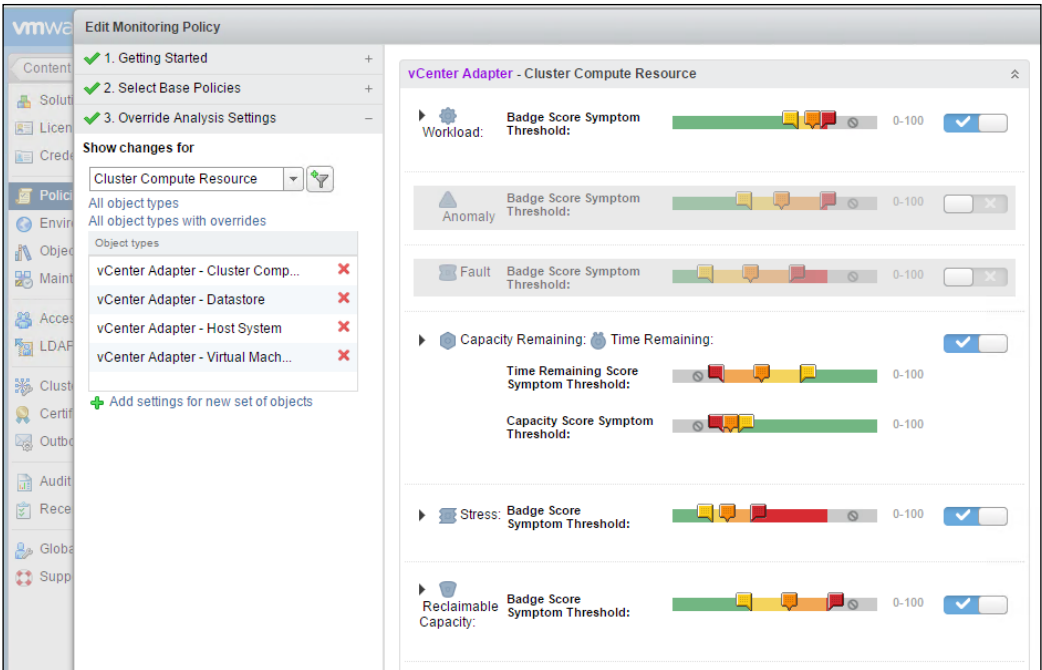
1. The first step to identifying idle VMs is to configure the default vROps policy to monitor them. Login to the vROps web portal as an administrator and navigate to the **Administration** section.

- Next, click on **Policies**. Take a look at the following screenshot as an example:



- After identifying the active policy, click on the **Policy Library** tab.
- Search for the active policy and click on **pencil** icon to edit it. A new pop-up screen will appear.

5. Under **Override Analysis Settings**, enable **Reclaimable Capacity** under the various components in the center of the screen. Refer to the following screenshot for an example and hit the **Save** button in the bottom right-hand corner when you've finished:



6. Next, we need to run a report in vROps to see which VMs are listed as idle. In the upper left-hand corner of the vROps web portal, click on the **Content** icon. Once this screen has loaded, click on **Reports**, which is on the left-hand side.

7. In the center workspace, under **Report Templates**, search for the **Idle VMs Report**. Click on it once and then click on the **Run Template** icon:

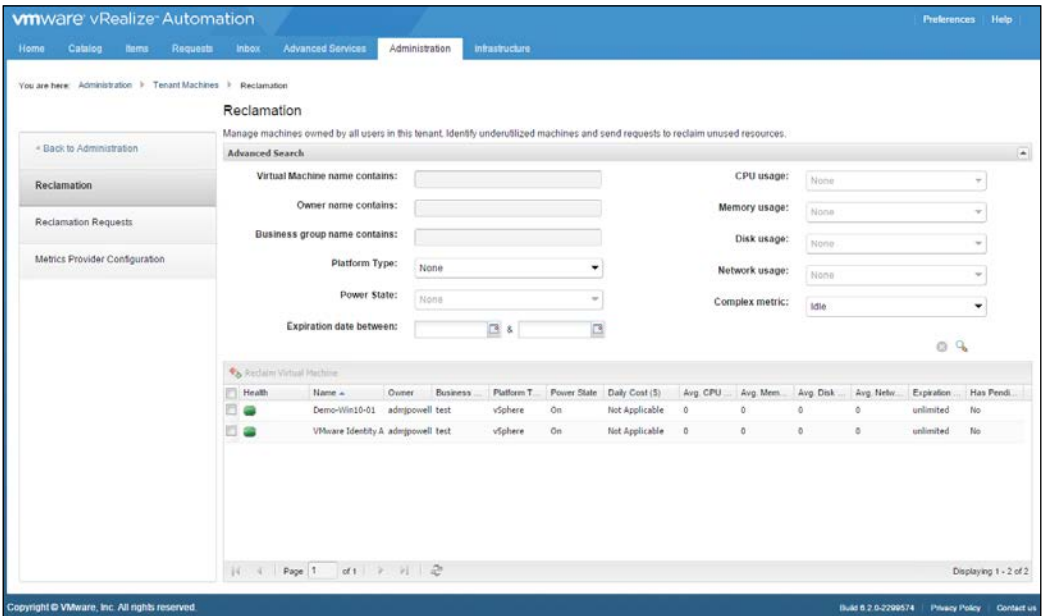
The screenshot shows the VMware vRealize Operations Manager interface. The left sidebar contains navigation options: Home, Dashboards, Views, Reports (selected), Alert Definitions, Symptom Definitions, Actions, Recommendations, Notifications, Super Metrics, Group Types, and Icons. The main workspace is titled 'Reports' and has two tabs: 'Report Templates' and 'Generated Reports'. The 'Report Templates' tab is active, displaying a table of report templates. The 'Idle VMs Report' is highlighted in blue at the bottom of the list.

Name	Subject
<b>Datastore Min/Max/Avg IOPs 30 Days List View Report</b> Generated reports (0)	Datastore
<b>Datastore Used Space (%) Distribution Report</b> Generated reports (0)	Datastore
<b>Datastore Waste List View Report</b> Generated reports (0)	Datastore
<b>Host CPU Demand (%) Distribution Report</b> Generated reports (0)	Host System
<b>Host CPU Demand (%) Trend View Report</b> Generated reports (0)	Host System
<b>Host Hardware Summary</b> Generated reports (0)	Host System
<b>Host Important Services Summary</b> Generated reports (0)	Host System
<b>Host Memory Usage (%) Distribution Report</b> Generated reports (0)	Host System
<b>Host Memory Usage and Demand (%) Trend View Report</b> Generated reports (0)	Cluster Compute Resource
<b>Host Status Summary</b> Generated reports (0)	Host System
<b>Idle VMs Report</b> Generated reports (1)	Virtual Machine

8. Running the report will take several minutes, depending on the size of your environment. Once this is complete, review the machines listed in the report.

9. Now that we know which machines are idle in our environment, we will want to compare them to the ones listed in the vRA web portal. Log back into the portal and go to the **Administration** tab, click on **Tenant Machines**, and then on **Reclamation**. Clicking on the down arrow, located on the right-hand side under **Advanced Search**. This will yield a variety of options.

With regard to searching for idle VMs, we will want to change **Complex metric** from its default of **None** to **Idle**, as shown in the following screenshot:



10. Clicking on the **magnifying glass** icon will search for idle VMs in the purview of vRA. If idle VMs are located, they will be listed at the bottom of the screen.
11. Free up compute resources by removing one of the idle VMs listed. As the preceding screenshot shows you, we will remove **Demo-Win10-01**. To do this, put a check mark in the far left-hand side of the VM. Now, click on the **Reclaim Virtual Machine** button in the left-hand corner listed above the idle VMs.
12. You can monitor these requests by navigating to **Reclamation Requests**, located in the navigation menu on the left-hand side of the vRealize Automation web portal.

## Summary

In this chapter, we provided an overview of vRealize Operations Manager 6.0, as well as the benefits it provides to vRealize Automation. Through the chapter, we detailed the steps necessary to integrate vROps with vRA. We also showed how it is possible to decommission idle VMs, by having them identified through vROps. This is beneficial, as it returns consumed and unused resources back to the vRealize Automation compute pool. This is a great tool that provides intelligent control over the total compute resources in a vRA environment.

In the next chapter, we are going to explore the vRealize Automation web portal through the eyes of your customer, the end user. We will cover how to customize the web portal to give the end user the information they need to make their experience as valuable as possible. This will include navigating through the web portal, customizing the home page, and making service requests for VMs, applications, and services.



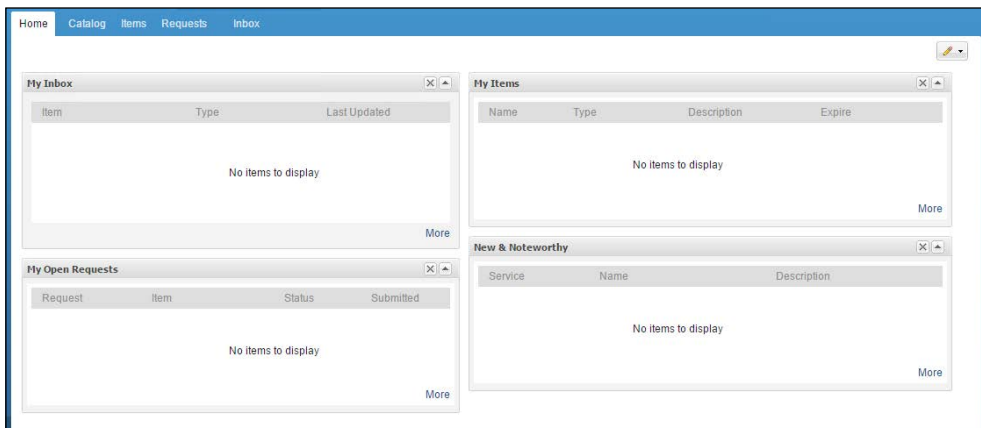
# 10

## Customizing the End User Portal Experience

In this chapter, we are going to explore the vRealize Automation web portal from the perspective of an end user. We will customize the features of the main page of the portal, as well as make the overall look of the portal fit our company's image. Finally, we will go over some of the ways to request and leverage services and end user provisions through the vRA web portal.

### Portal overview

If you have been following the examples in this book, you are likely comfortable navigating around the vRA web portal, through an administrator's perspective. Let's step back and take a fresh look at the portal through the eyes of the end user. When an end user logs into the vRealize Automation web portal for the first time, they are greeted with a screen similar to the following:

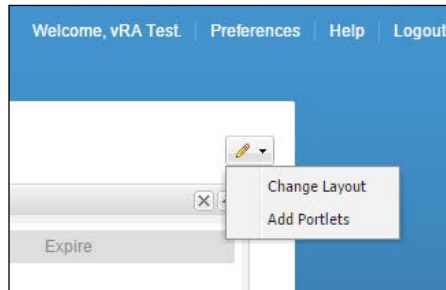




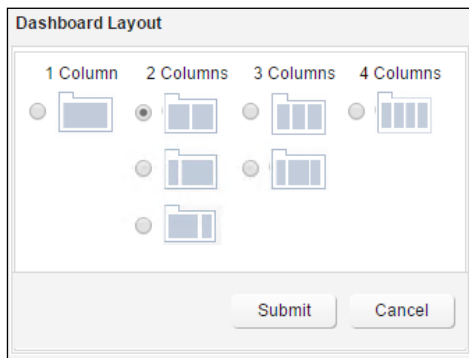
This screenshot represents an out-of-the-box experience and is a blank canvas. The main page is broken into four quadrants. They are outlined below, starting from the upper left-hand corner and moving to the right:

- **My Inbox:** This section lists any requests that may be pending approval from managers or other service approvers we defined
- **My Items:** This is a list of the various VMs and applications provided to the end user and are available for them to use
- **My Open Requests:** This section lists all the service requests submitted by the end user. Any items with a pending approval status will appear in this area
- **New and Noteworthy:** This section is for catalog items that the administrator has flagged as **New and Noteworthy**. It is configured in the **Catalog** and **Items** tab

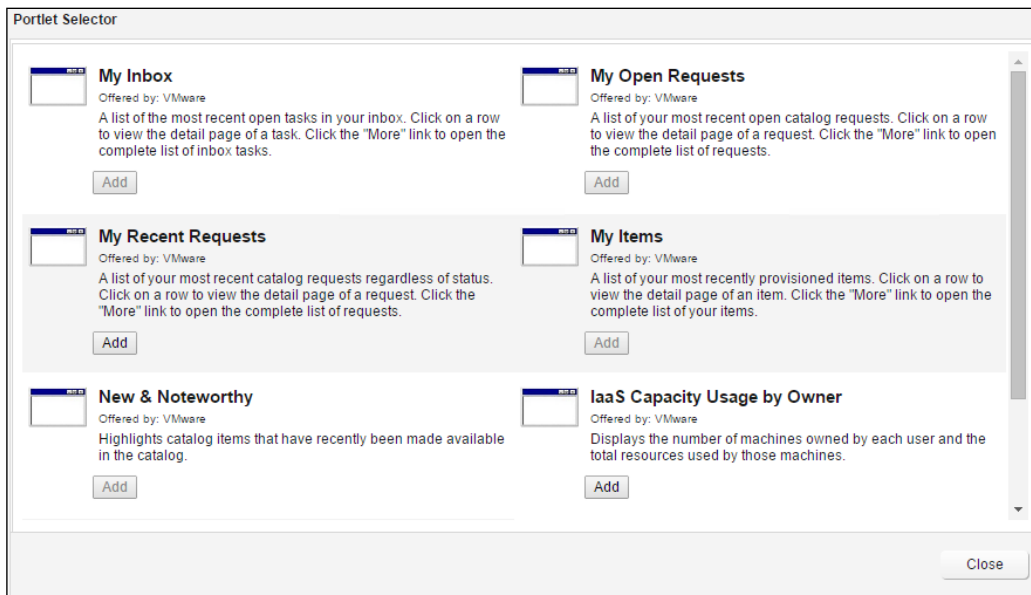
In the upper right-hand corner, there is a **pencil** icon just below the end user's login name. Clicking on the icon will reveal a drop down menu, as shown in the following screenshot:



1. Clicking on **Change Layout** will display a pop-up dialog box, similar to this one:

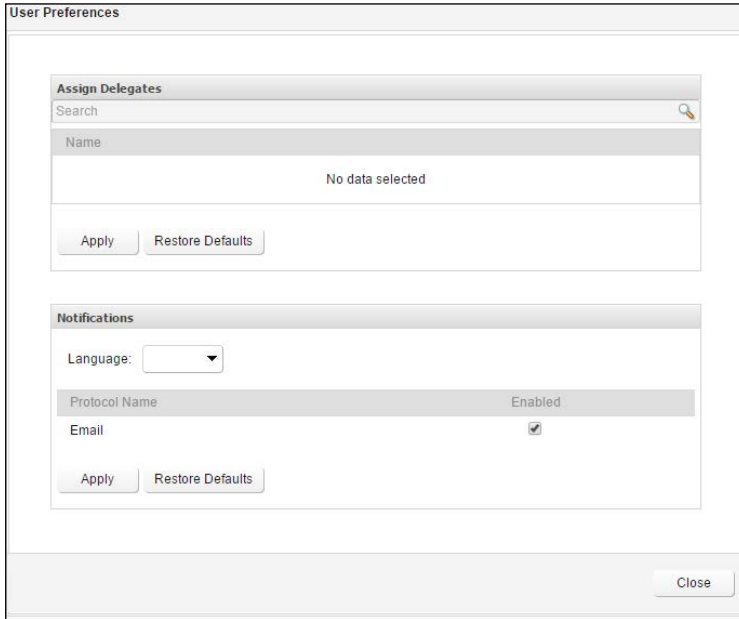


2. The **Dashboard Layout** allows the end user (and administrators) to change the number of columns and their shapes for the main vRealize Automation page. After making any changes, you can hit the **Submit** button to make these changes take effect.
3. Going back to the drop-down menu and clicking on **Add Portlets** will yield this dialog box:



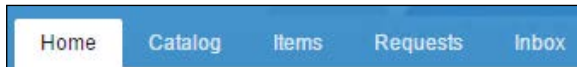
4. This can be used to populate the main vRA page with additional items and views. Simply click on the **Add** button below the portlets you wish to add, and then hit **Close**.
5. Once you're back on the main page, you can drag and drop the various portlets to customize the workspace even further.

6. Clicking on the **Preferences** link to the right of your logged in username displays this following dialog box:



7. End users can assign delegates or choose whether or not to receive notifications. Once the changes are modified, click on the **Close** button.

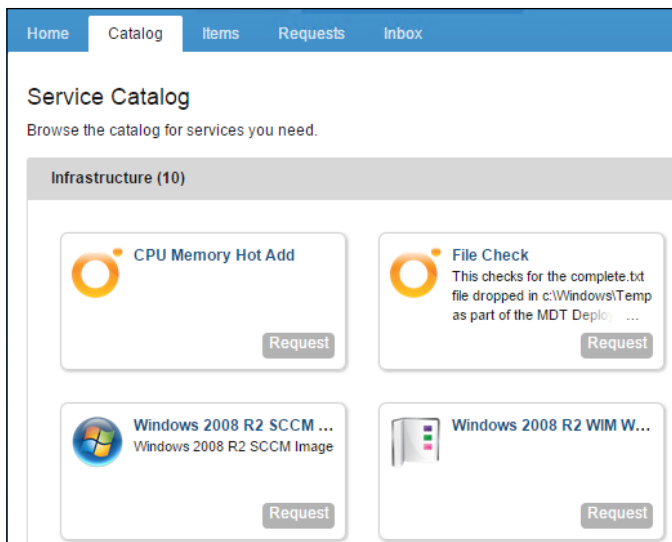
Notice on the main page there are fewer tabs than you saw earlier when you logged in as a vRA administrator. See below for an example:



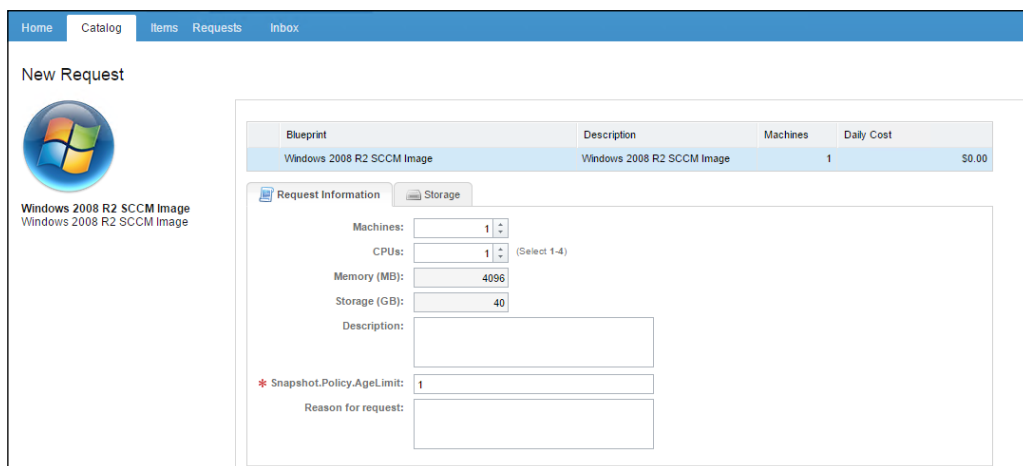
The preceding tabs represent the areas in which an end user can navigate. Now, let's take a look at requesting infrastructure.

## How to make service requests

As an end user, click on the **Catalog** tab. This will list the various services that you have been entitled to use. Here is an example that shows two operating systems and the vRO workflows that we created and published in *Chapter 8, REST API and vRealize Orchestrator*:

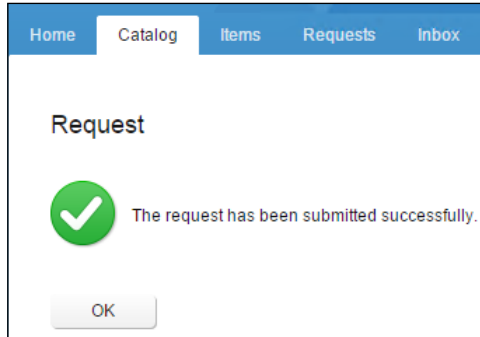


Clicking on the **Request** button, as shown in the following screenshot, will show that each of the items will yield different results. This is based on a number of factors, but it boils down to the various options and prompts the administrator during the creation of the Blueprints. The request shown here represents a basic Windows Blueprint:

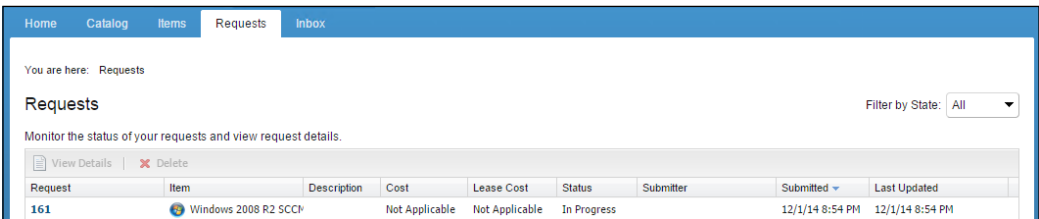


There are parameters that can be adjusted in this service request, such as **Machines**, **CPU**, **Memory**, and **Storage**. Depending on additional configurations by the administrator, there may be approval policies tied to your request.

However, if you take the defaults presented and hit **Submit**, your request will be processed and you will be redirected to this screen:



You can monitor the status of your request by going back to the main page, or clicking on the **Requests** tab:



The end user will be able to monitor their progress. It is likely that the administrator has configured notifications, and the end user will receive an e-mail from the vRA web portal, which lets them know that they can use the VM they've provisioned.

## How to monitor service requests

As we alluded to in the preceding section, the end user can monitor the status of their request in several ways:

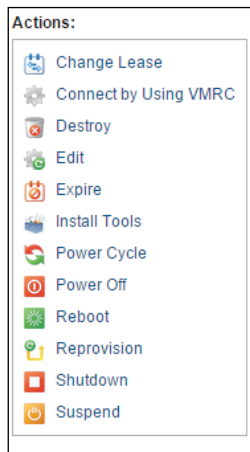
- On the main page of the vRealize Automation web portal under the **My Open Requests** workspace
- Under the **Requests** tab. Clicking on the item number of the request will reveal additional details about the status of the request.
- Under the **Inbox** tab. Clicking here will display any requests with pending approvals. You can also see if an approval was accepted or rejected. This is also located on the main page of the vRA web portal by default.

- Waiting for e-mail. Assuming your end users are busy, this is the most likely way an they will know when their service request is fulfilled. After all, software developers usually stay quite busy.

## Installing plugins to access VMs through the web portal

One of the most powerful features is also one of its most simple: Providing access to the applications and infrastructure the end user requested directly from the portal. Using this feature prevents the need to open additional firewall rules, as the ports being used are accessed over the vRA web portal's ports. However, the web browser cannot access these services without installing additional plugins.

Take for example, a Windows VM. Once provisioned, we can click on the **Items** tab and access the VM. After clicking on the provisioned VM, you'll see an **Actions** menu on the left-hand side, as illustrated here:



We will go over the **Action** items in the next section. For this example, we want to concentrate on **Connect by Using VMRC (VMware Remote Control)**. This will pop-up a new tab or window, depending on your web browser. At the top of the new window will be a link to download and install the VMRC plugin. Be sure to do so and you will be able to access the GUI of any Windows or Linux system.

Although it's not displayed in the preceding screenshot, if you provisioned a Linux OS and selected **Custom PropertyMachine, SSH** with a **Value** of **True**, you will see an **SSH** option. This will allow you to connect via an **SSH** session from your browser.

As exciting as this may be for the end user, it does involve a bit of customization on their end point. SSH is not a native application to Windows, and it would need an SSH client. Additionally, you would have to configure the SSH client to respond to SSH://calls in a web browser. If you are using Chrome or Firefox, you have several choices in the form of plugins. It is a bit more difficult to find support for Internet Explorer or Microsoft Edge. One such program is KiTTY. It is essentially Putty, but it allows for SSH://calls after importing registry keys in Windows. Additionally, you can find a customized version of Putty at <http://dailyhypervisor.com/download/>, which is a great resource for vRealize Automation information.

## Modify previously deployed service requests

How often does a business unit come back to the infrastructure or application team and request additional resources? It happens quite often in almost every company. As we have already learned, vRealize Automation allows us to provide a range of options for the items we entitle to the end user. If a need arises from the software development team, they can simply log back into the vRA web portal, click on the **Items** tab, and select the service they want to modify. On the **Actions** tab, we have the following options and their purpose:

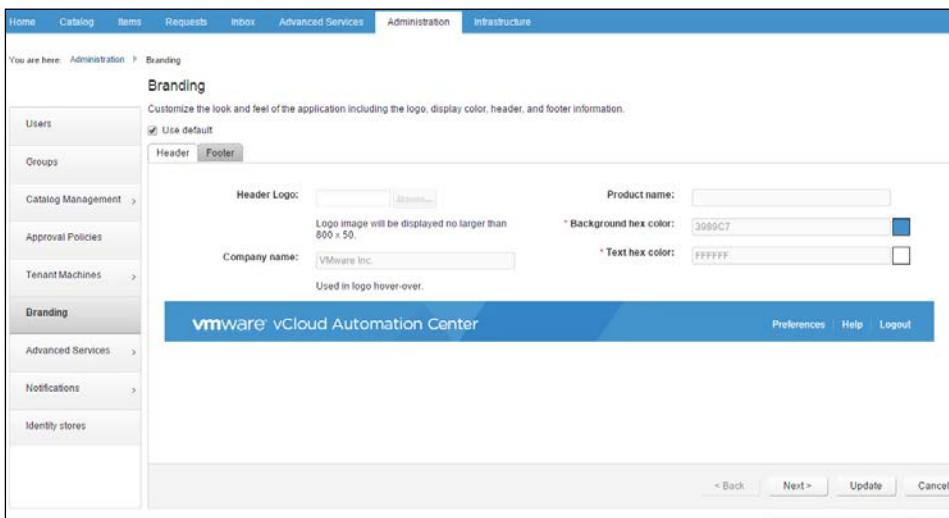
- **Change Lease:** Change the amount of time that a VM will be used. This is normally defined by the vRA administrators and can be extended up to a period of time, if allowed.
- **Connect by Using VMRC:** Connect to an operating system's GUI.
- **Connect by SSH:** It is available if defined in the **Build Profile** or **Properties** of a VM. It's only used by Linux-based systems.
- **Destroy:** Delete all traces of a VM, including unregistering it from the **Hypervisor** and removing it from disk.
- **Reconfigure:** Modify the CPU, memory, and disk using this option.
- **Expire:** Depending on the situation, but normally to expire a machine means to destroy it. The exception is when an archive period has been defined. In this case, it is powered off and stored for the duration of the archive period. Note, this is calculated based on when the service item is requested, not when it was provisioned.
- **Install Tools:** Install VMware tools.
- **Power Cycle:** Powers off the machine completely, then powers it back on automatically.

- **Power off:** Powers off the machine and makes it inaccessible until it is powered on again manually.
- **Reboot:** This is self-explanatory.
- **Reprovision:** This will make a new request for the exact same item.
- **Shutdown:** This is self-explanatory.
- **Suspend:** Pauses the state of the VM.

Once an end user modifies a previously deployed service request, they can monitor the request in the same methods we outlined at the beginning of the chapter.

## Customizing the end user web portal

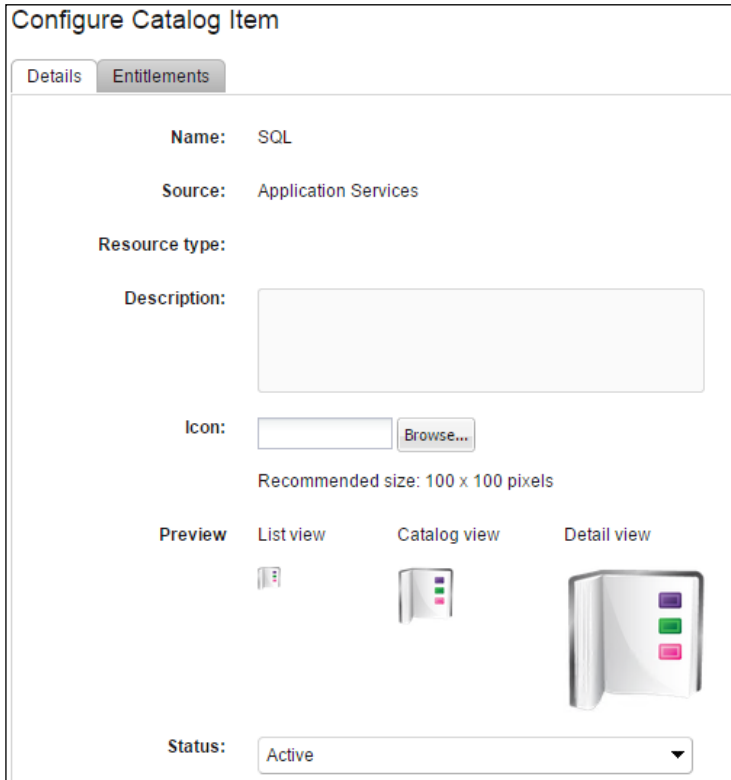
VMware has given us customization options for the vRA web portal. VMware allows us to customize the header and footer, as well as the background color of the vRA web portal. These options are found under the **Administration** tab, and then **Branding**:



Additionally, you can customize each catalog item by providing a custom icon. vRealize Automation ships with a limited amount of icons, excluding many of the more noticeable icons that end users would recognize, such as operating systems by Microsoft, Oracle, and Red Hat. You can create your own, or download a customized icon pack from the Internet. When you are ready to change an icon, simply click on the **Administration** tab, then on **Catalog Items**. Select the **Catalog Items** you want to edit and hit the **Browse** button. Select the icon you wish to represent your item, then click on the **Update** button.



Take a look at the following screenshot:



## Summary

In this chapter, we covered the end user web portal from the perspective of the end user. We learned how a user can request services and access them after installing the proper plugins.

From the vRealize Automation administrative perspective, we learned how to customize the web portal for an individualized company experience.

In the next chapter, we will explore how to troubleshoot the components that comprise the vRealize Automation 6.2 universe.

# 11

## Troubleshooting vRealize Automation 6.2

In this chapter, we are going to investigate some of the common errors that you may encounter during the various phases of utilizing vRealize Automation. We will explore the following topics in the chapter:

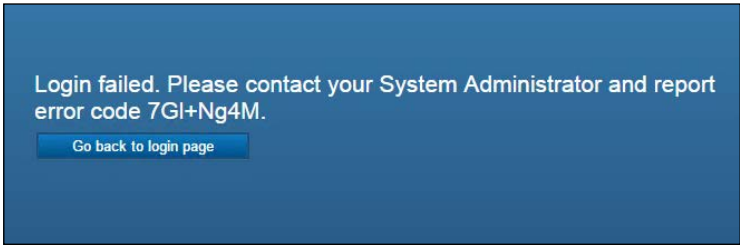
- Errors and how to decode their meaning
- Various monitoring sources and log files
- Getting further assistance

### **Errors that may occur during installation and operation**

Hitting roadblocks is common when installing and familiarizing ourselves with new technologies. With vRA, there are a multitude of errors that you could encounter. If you have been following the detailed steps provided throughout the book, you likely avoided the pitfalls of getting the environment up and running.

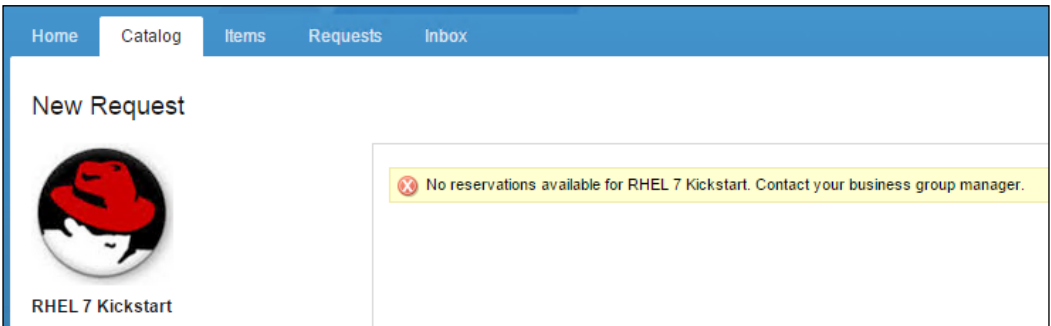
As a general troubleshooting rule, try using a template or snapshot to do your first round of testing and troubleshooting. It is the most basic way to deploy infrastructure and services through vRA. It will also provide focus for your troubleshooting efforts. For example, if an AppD service deploys correctly from a template, but not from SCCM or WIM, you will need to focus on the difference between the three in terms of deploying the AppD Bootstrap agent. Perhaps you have been scripting the AppD Bootstrap agent, and this is the source of your problem.

The following screenshots show some of the most common errors, and how to overcome them so you can concentrate on supporting and developing your new vRealize Automation environment:



**Login failed:** The error code shown in the preceding screenshot, may seem useful, but actually is not. This is because the time between the vRA Appliance and Identity Server has drifted too far apart or you have an error with the certificates you are using. Login to both of the appliances web interfaces and adjust the time and check the certificates. On rare occasions, you may need to *ssh* into the appliances and set time from the command line. Also, keeping both servers on the same ESXi host will keep time in sync if you use the hosts as the time source.

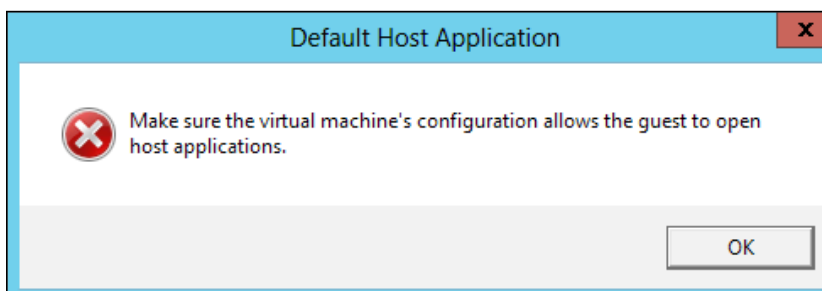
If you are already inside the portal while the time has drifted, this error may prevent you from executing services successfully, such as decommissioning a VM or requesting new ones. These are the most common issues and they cause lots of problems. If you see a white page once you logged in, it is also a symptom of time drifting. It is recommended to check the time sync as a first step in the troubleshooting process.



**No reservations available for <catalog item>. Contact your business group manager:** This can occur in two different scenarios: either when there is no reservation assigned or there are no resources available.

In the **Infrastructure** tab, click on **Reservations**, and then on **Reservations** once more. If you only have one reservation present, make sure there are enough resources to create a new VM. Typically, you will run out of memory when provisioning a machine instead of CPU. If you run out of memory in the reservation, add more and have the end user try again.

It is also possible that the end user is in a business group that does not have a resource group associated with it. If this is the case, either create a new reservation and associate it with the troubled user's business group, or associate an existing reservation to it.



**Make sure the virtual machine's configuration allows the guest to open host applications:** This can be corrected by adjusting the file association in Windows. The error occurs when the **Virtual Machine Remote Control (VMRC)** plugin we installed earlier, is trying to launch SSH sessions instead of the SSH program that is installed. In Windows, this can be adjusted in **Control Panel** under **Default Programs**.

**OOB sample content not copied when setting up vRealize Application Services or vRA Applications Server homepage is blank upon logging in:** Login to the AppD vApp via SSH with the `darwin_user` user. Once logged in, `su` to elevate your privileges.

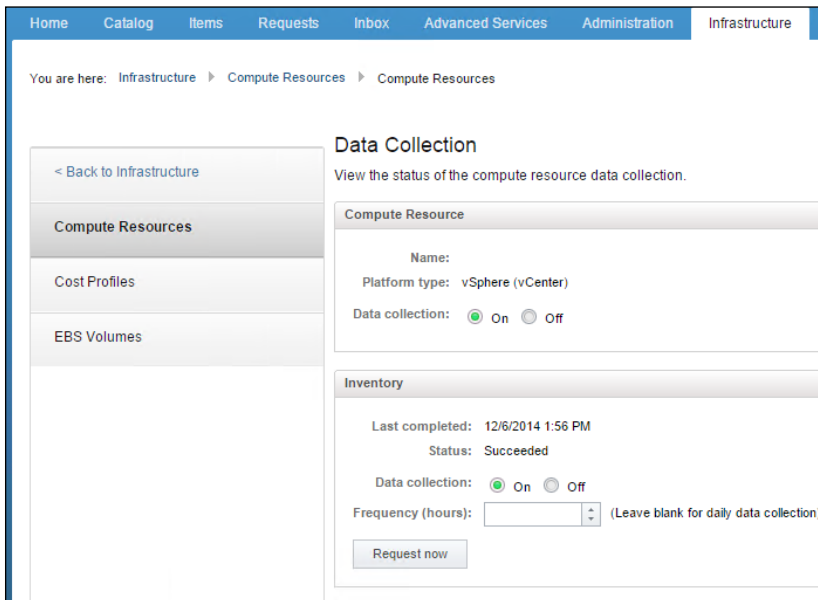
Go to `/home/darwin/tools` and rerun the `import_oob_content.sh` script and follow the prompts.

**vRO workflows do not run when published to vRA as machine custom properties:** There are two things that come into play. If you have upgraded to vRA 6.2 from 6.0 or 6.1, there is an error with Java on the vRA Appliance. Following the VMware **Knowledge Base (KB)** resolves this issue: [http://kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayKC&externalId=2093444](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2093444).

If the vRO workflows published to vRA never log their attempts to run in the vRO server, you will need to run the **Install vCO customization** workflow. It is located in Library | vCloud Automation Center | Extensibility | Administration | Installation. Select your IaaS server. Select **Yes** for all options under **State change workflow stubs to update to run vCO workflow** and change **Number of menu operations and their workflows** from 8 . 0 to 0 before running it.

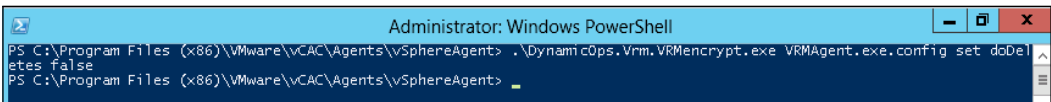
**Newly created vCenter template does not appear in vRA:** By default, the vRA inventory service only runs once per day. In order to see newly created templates, you must run inventory service manually. To do this, login to the vRA web instance by following these steps:

1. Go to the **Inventory** tab, then click on **Compute Resources**.
2. Click on **Compute Resources** one more time and you will see the resources in the main workspace.
3. Hover your mouse over the compute resource and a drop-down menu will appear. Select **Data collection**.
4. In the **Inventory** section, hit the **Request Now** button. It could take 5-10 minutes to complete this action, based on the size of your environment. You also can change the default inventory frequency by modifying the **Frequency (hours)** field. Take a look at this example:



**VMs are destroyed before they are presented for use:** While this is rather generic, it is quite annoying if you're not familiar with deploying from Kickstart, SCCM, or WIM. Normally, two issues are at play here:

- **The VM is not getting proper network connectivity:** This could be due to the DHCP server running out of available addresses. If you are working on Windows, you can hit *F8* and issue the `ipconfig` command. If a `169.x.x.x` address is returned, it is likely that you have not received a valid IP address. However, if you are savvy, you can manually add one via the command line and probably continue without a reboot.
- **The vRA agent is not installed or deployed correctly:** This scenario is less likely if you install the agent in a **golden image** or the template that you use to deploy. If you've pushed it out after the OS is installed, such as via SCCM, this could be the issue. If the IaaS server does not receive a signal from the agent on the deployed OS, the IaaS server will destroy the VM after its time out. In order to prevent this from occurring, you can login to the IaaS server and make the change shown in the following screenshot via PowerShell:



```
Administrator: Windows PowerShell
PS C:\Program Files (x86)\VMware\VCAC\Agents\vsphereAgent> .\DynamicOps.Vm.VRMencrypt.exe VRMAgent.exe.config set doDelete false
PS C:\Program Files (x86)\VMware\VCAC\Agents\vsphereAgent> _
```

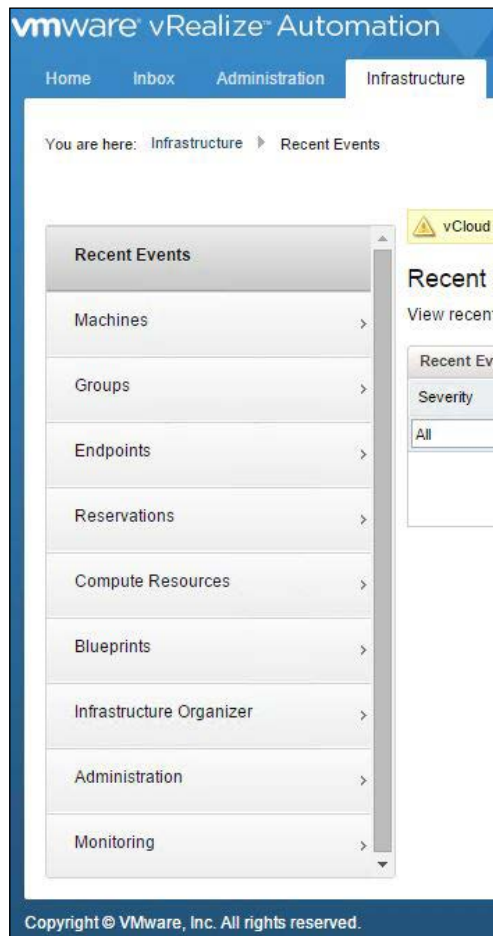
Once this is complete, restart the **vCloud Automation Center Agent - agentname** service from the **Services Microsoft Management Console (MMC)**. You can set this back to `True` if you wish, by repeating the process.

**Applications in AppD fail or do not deploy as expected:** There are several places to check in order to see the point of failure when you deploy applications from AppD. Follow these guidelines to help in the troubleshooting process:

- **Perform the AppD deployment from the AppD web portal:** This will allow you to see exactly what point in the provisioning process the failure occurs. This will give you a focal point for troubleshooting. After deploying the application from AppD, select **Deployments** from the drop-down menu on the right-hand side of the portal. Select the application, then zoom in on **Execution Plan**.
- **Check the AppD Logs:** The logs are located in this directory of Windows: `c:\opt\vmware-appdirector\agent\logs`.
- **Ensure VMware tools are installed and running:** Without VMware tools running, AppD will not be able to get the insight it needs into the OS to deploy services or applications.

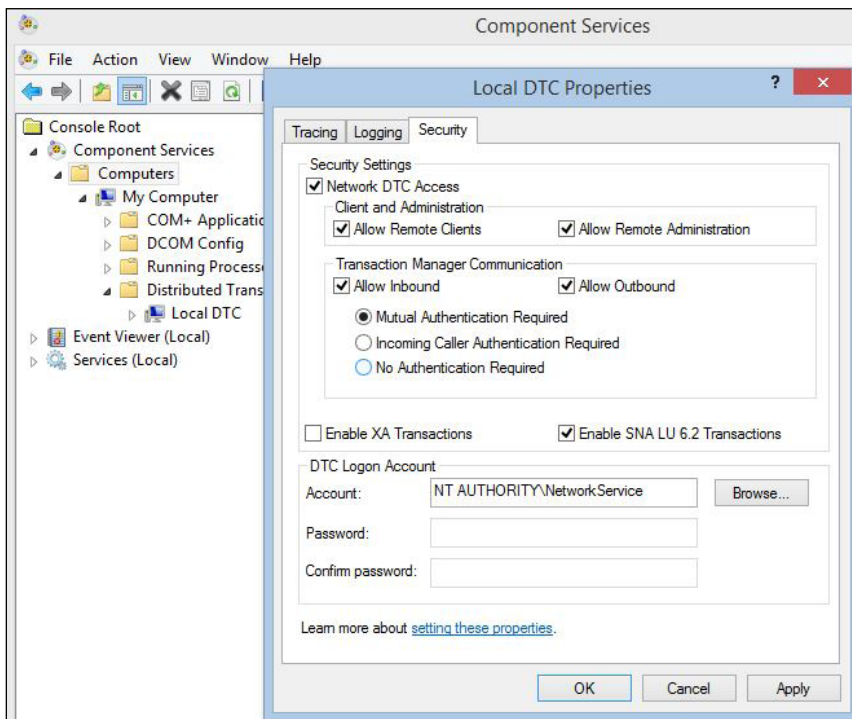
- **Make sure there are no older versions of the AppD or vRA agents installed:** If you are upgrading from a previous version, you must uninstall the previous agent services and remove their metadata.
- **Ensure your network works on the deployed VM:** The guest VM must be able to access the AppD portal to get additional files and information.

**Missing links under the Infrastructure tab:** When the vRA Appliance, Identity, and IaaS server have been configured correctly (as covered in *Chapter 4, Installing and Configuring vRealize Automation 6.2*), you should see the following links available under the **Infrastructure** tab:



However, it is not uncommon to see an incomplete list. This can be caused due to several factors:

- **Your IaaS services and SQL database are cloned or built from the same vCenter template / Your IaaS services and SQL database are located on separate VMs:** This would be a common scenario for a production environment. The issue has to do with **MSDTC** (short for **Microsoft Distributed Transaction Coordinator**) permissions. Perform these steps to fix this issue:
  1. On the IaaS server, open an elevated Command Prompt.
  2. Type `msdtc -uninstall` and hit *Enter*.
  3. Type `msdtc -install` and hit *Enter*.
  4. **Open Component Services.** Expand Component Services | Computers | My Computer | Distributed Transaction Coordinator. You will see **Local DTC**. Right-click on it and select **Properties**. Click on the **Security** tab; deselect all options except **Enable SNA LU 6.2 Transactions**. Take a look at the following screenshot:





5. Open the **Services** snap-in. Restart all the vRA related services.
  6. Repeat this process on the SQL Server.
  7. Log back into the vRA web portal (if you've already logged in, log out and then back in). You should see all the links under the **Infrastructure** tab.
- **There is a name mismatch between the vRA web portal endpoint and the IaaS server:** In the vRA web portal under Infrastructure | Monitoring | Log look for a log that occurs every minute with the message "*This exception was caught*": *The attached endpoint 'vCenter' cannot be found*. Whatever is in single quotes (in this example, vCenter) is what the IaaS server is looking for in the vRA web portal as the name of an endpoint. Check your endpoint and make sure it is named exactly as listed in the error.

## Monitoring and logging

Since there are so many components that make up the vRA universe, it makes sense that there would be numerous places to check for monitoring and logging. In the vRA web portal, under the **Infrastructure** tab, there is a link for **Monitoring**. **Monitoring** has four sub categories, which are listed here:

- **Audit Log:** This is where you can find which user provisioned a VM, requested more resources, or decommissioned a VM, among other things.
- **Distributed Execution Status:** This provides insight into the IaaS server where the DEM and Orchestrator reside. You can get details on each, as well as the workflows that are being executed. Also, you can check connectivity to the IaaS server, as these two items need to be listed as **Online** in order for them to execute workflows properly.
- **Log:** This contains a detailed and historical log for all the events that pass through the vRA web portal. This is a great place to begin troubleshooting.
- **Workflow History:** This should be an obvious one, as it contains all of the events tied to the workflows executed within the vRA web portal.

It is worth noting that each vApp has its own dedicated logs, and the IaaS server will log events to the Windows Event Viewer, as well `C:\Program Files(x86)\VMware\VCAC\Agents\Logs\vsphereAgent.log`. For the vApps, the catalina logs provide a wealth of information that will help you troubleshoot issues. You can find these logs at `/var/log/vcac` in the vRA Appliance.

Digging through these logs will give you a greater understanding of issues in your environment. Also, you can generate a support bundle. This is normally requested when you open a case with VMware support, but you can also gather all the logs in a centralized ZIP file for your own troubleshooting needs.

## Getting further assistance

There are a number of places on the web where you can get additional assistance. The first stop would be the VMware Community forums. You will have to register to use the forum, which is a free process. You can use the same account that you utilized to download the software in the first place. The community forum has a wealth of information and VMware employees often contribute to it and answer questions.

Your company has invested a great deal of money in this product, and it is certainly encouraged to engage VMware Professional Services and support when you need immediate help. In the next chapter, we will provide resources to help you learn more about vRA as well as other great support sites.

## Summary

This chapter covered some of the more common errors and problems associated with installing, configuring and running the various components of the vRealize Automation universe. Due to the complexity of this product, it is always recommended to engage VMware support or professional services for more troubleshooting and configuration challenges.

In *Chapter 12, References for vRealize Automation 6.2*, the final chapter, we will provide a list of resources you can utilize for a greater understanding of vRA as you continue your journey of mastering vRealize Automation 6.2.



# 12

## References for vRealize Automation 6.2

This final chapter services more as a link of references and less of an explanation of concepts. After all, you made it to the end, and I am sure you are ready for some outside perspective. Please note that vRealize Automation 6.2 has been on the market for less than a year at the time of writing this book. Therefore, there is not an abundance of reference material out there specific to 6.2. However, there is a wealth of material that describes the features of series 6.0 of this product.

### Links within your vRA environment

The following are some of the common links built into vRealize Automation 6.2. We will continue to use the `domain.local` format that we've used throughout this book, in the links listed for this section:

- vRealize Automation 6.2 user portal login page at `https://vRA.domain.local/vcac`
- vRealize Automation software downloads at `https://vRA.domain.local`
- IaaS Installer components at `https://vRA.domain.local:5480/installer`
- Individual vRA component status at `https://vRA.domain.local/component-registry/services/status/current`
- vRealize Application services login at `https://vRAappd.domain.local:8443/darwin/org/vsphere.local`

## VMware documentation

The following are some more useful links to help you with vRealize Automation 6.2:

- IaaS configuration for Virtual Platforms: VMware Realize Automation 6.2 at <http://pubs.vmware.com/vra-62/topic/com.vmware.ICbase/PDF/vrealize-automation-62-iaas-configuration-for-virtual-platforms.pdf>
- VMware Realize Automation 6.2 release notes at <https://www.vmware.com/support/vcac/doc/vrealize-automation-62-release-notes.html>
- VMware vCloud 6.1 Automation Center Reference Architecture at <http://www.vmware.com/files/pdf/products/vCloud/VMware-vCloud-Automation-Center-61-Reference-Architecture.pdf>
- VMware vRealize FAQ at <http://www.vmware.com/files/pdf/products/vrealize-cmp/vmw-faq-vrealize-rebrand.pdf>
- vRealize Automation home page at <http://www.vmware.com/products/vrealize-automation/>
- VMware Hands on Lab: HOL-SDC-1421 - Brokering IT Services and Applications with vCloud Automation Center at <http://labs.hol.vmware.com/HOL/catalogs/>
- VMware plugins for vRA at <https://solutionexchange.vmware.com/store>
- VMware Guest Script Manager Package at <https://communities.vmware.com/docs/DOC-25474>
- VMware vRealize Orchestrator learning at <http://www.vmwarelearning.com/orchestrator/>
- VMware vRealize Application Services Documentation at <http://pubs.vmware.com/vCAC-61/topic/com.vmware.ICbase/PDF/vcloud-automation-center-61-using-application-services.pdf>
- VMware Onyx Project at <https://communities.vmware.com/community/vmtn/automationtools/onyx>
- VMware customized property list for blueprints at <http://pubs.vmware.com/vCAC-60/topic/com.vmware.ICbase/PDF/vcloud-automation-center-60-custom-property-reference.pdf>

The following chart lists the vRealize Automation custom properties organized by property name:

Custom properties table type	Property	Description
A	AD.Lookup.Department	Specifies the cost center value that is included in e-mail sent to approvers to notify them of pending approvals.
A	Amazon.ElasticLoadBalancer.Names	Assigns machines that are provisioned by a blueprint to the elastic load balancers that match the specified values.
A	Amazon.Instance.Id	Specifies the Amazon instance ID of a machine provisioned on an Amazon EC2 endpoint.
B	BMC.Service.Profile	Specifies the name of the default authentication profile on the BMC BladeLogic server.
B	BMC.AddServer.Delay	Specifies the number of seconds to wait before adding the machine to BMC BladeLogic. The default is 30.
B	BMC.AddServer.Retry	Specifies the number of seconds to wait before retrying if the first attempt to add the machine to BMC BladeLogic Configuration Manager is unsuccessful. The default is 100.
B	BMC.Software.Install	Set to <code>True</code> to enable BMC BladeLogic integration
B	BMC.Software.BatchLocation	Specifies the location in BMC BladeLogic of software jobs to be deployed. This value must match either the appropriate field in <code>Website\Software.txt</code> or the appropriate value of <code>VRM.Software.IdNNNN</code> , depending on the method used to prepare software jobs for integration.
C	Cisco.Organization.DN	The distinguished name of the Cisco UCS Manager organization in which Cisco USC machines provisioned by the provisioning group are placed, for example <code>org-root/org-Engineering</code> . If the specified organization does not exist in the Cisco UCS Manager instance managing the machine, provisioning fails. This property is available for provisioning groups only.
C	CloneFrom	Specifies the name of an existing machine or virtualization platform object to clone from, for example a template in vCenter Server. The default is the value specified by the <b>Clone from</b> setting on the blueprint's <b>Build Information</b> tab.

Custom properties table type	Property	Description
C	CloneSpec	Specifies a cloned machine, for example a predefined SysPrep object in vCenter Server. The default is the value specified by the customization setting on the blueprint's <b>Build Information</b> tab.
C	Command.DiskPart.Options	When you use WIM-based virtual provisioning on ESX server hosts, set to <code>Align=64</code> to use the recommended alignment parameters when you format and partition the machine's disk. This property is not available for physical provisioning.
C	Command.FormatDisk.Options	When you use WIM-based virtual provisioning on ESX server hosts, set to <code>/A:32K</code> to use the recommended alignment parameters when you format and partition the machine's disk. This property is not available for physical provisioning.
E	EPI.Server.Name	Specifies the name of the external provisioning infrastructure server, for example, the name of the server hosting BMC BladeLogic. If at least one general BMC EPI agent was installed without specifying a BMC BladeLogic Configuration Manager host, this value directs the request to the desired server. If only dedicated BMC EPI agents for specific BMC BladeLogic Configuration Manager hosts were installed, this value must exactly match the server name configured for one of these agents. Specifies the name of the server hosting HP server automation. If at least one general Opsware EPI agent was installed without specifying a server automation server, this value directs the request to the desired server. If only dedicated Opsware EPI agents for specific HP server automation servers were installed, this value must exactly match the server name configured for one of these agents. If at least one general EPI agent of the appropriate type ( <code>VirtualMachine.EPI.Type</code> ) was installed without specifying a server, this value directs the request to the desired server. If only dedicated EPI agents for specific servers of the appropriate type were installed, this value must exactly match the server name configured for one of these agents.

Custom properties table type	Property	Description
E	<code>EPI.Server.Port</code>	Specifies the port on which to contact the provisioning server. If you are using a Citrix provisioning server, omit to specify the default port value of 54321.
E	<code>EPI.Server.VDiskName</code>	Specifies the name of the Citrix provisioning vDisk to provision from.
E	<code>EPI.Server.Store</code>	Specifies the name of the Citrix provisioning store that contains the vDisk identified by the <code>EPI.Server.VDiskName</code> property.
E	<code>EPI.Server.Collection</code>	Specifies the name of the Citrix provisioning collection to which the machine will be registered.
E	<code>EPI.Server.Site</code>	Specifies the name of the Citrix provisioning site that contains the collection and store identified by the <code>EPI.Server.Collection</code> and <code>EPI.Server.Store</code> properties.
H	<code>Hyperv.Network.Type</code>	<p>This property is valid for use with Hyper-V only. The following values are available:</p> <ul style="list-style-type: none"> <li>• Synthetic (default)</li> <li>• Legacy</li> </ul> <p>This value is not available with Windows XP or Windows Server 2003 64-bit guest operating systems.</p>
H	Hostname	Specifies the machine name, overriding the generated machine name contained in the <code>VirtualMachine.Admin.Name</code> property. If <code>Hostname</code> is not used, the <code>VirtualMachine.Admin.Name</code> value is used as the machine name.
I	<code>Image.ISO.Location</code>	Specifies the location of the ISO image to boot from. The format of this value depends on your platform. For details, see the documentation provided by your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning. For virtual provisioning with vCenter Server, this specifies the name of a datastore in the instance that will be accessible to the provisioning compute resource. For virtual provisioning with XenServer, this specifies the name of a storage repository. For physical provisioning, this specifies the HTTP URL of the web-accessible location of the image.



Custom properties table type	Property	Description
I	Image.ISO.Name	Specifies the name of the ISO image from which to boot. The format of this value depends on your platform. For details, see the documentation provided by your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning. For virtual provisioning with vCenter Server, this value specifies the path to the image in the datastore specified by Image.ISO.Location, for example /MyISOs /Microsoft/MSDN/win2003.iso. The value must use forward slashes and begin with a forward slash. For virtual provisioning with XenServer, this value specifies the name of the image in the storage repository specified by Image.ISO.Location. In virtual provisioning with Hyper-V, this value specifies the full local path to the image. For physical provisioning, this value specifies the file name of the image.
I	Image.ISO.UserName	Specifies the user name to access the CIFS share in the format username@domain. For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.
I	Image.ISO.Password	Specifies the password associated with the Image.ISO.UserName property. For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.
I	Image.WIM.Path	Specifies the UNC path to the WIM file from which an image is extracted during WIM-based provisioning, for example \\server\share\$
I	Image.WIM.Name	Specifies the name of the WIM file located by the Image.WIM.Path property.
I	Image.WIM.Index	Specifies the index used to extract the desired image from the WIM file.
I	Image.Network.User	Specifies the user name with which to map the WIM image path (Image.WIM.Path) to a network drive on the provisioned machine. This is typically a domain account with access to the network share.
I	Image.Network.Password	Specifies the password associated with the Image.Network.User property

Custom properties table type	Property	Description
I	<code>Image.Network.Letter</code>	Specifies the drive letter to which the WIM image path is mapped on the provisioned machine. The default value is K.
I	<code>Infrastructure.Admin.MachineObjectOU</code>	Specifies the organizational unit (OU) of the machine. When machines are placed in the required OU by the provisioning group OU setting, this property is not required.
I	<code>Infrastructure.ResourcePool.Name</code>	Specifies the resource pool to which the machine belongs, if any. The default is the value specified in the reservation from which the machine was provisioned.
I	<code>Infrastructure.Admin.DefaultDomain</code>	Specifies the default domain on the machine
I	<code>Infrastructure.Admin.ADUser</code>	Specifies the user name that the machine uses to query Active Directory users and groups when an anonymous bind cannot be used.
I	<code>Infrastructure.Admin.ADPassWord</code>	Specifies the password associated with the <code>Infrastructure.Admin.ADUser</code> user name
L	<code>Linux.ExternalScript.Name</code>	Specifies the name of a customization script that the Linux agent runs after the operating system is installed. This property is available for Linux machines cloned from templates on which the Linux agent is installed. If you specify an external script, you must also define its location by using the <code>Linux.ExternalScript.LocationType</code> and <code>Linux.ExternalScript.Path</code> properties.
L	<code>Linux.ExternalScript.LocationType</code>	Specifies the location type of the customization script named in the <code>Linux.ExternalScript.Name</code> property. This can be either <code>local</code> or <code>nfs</code> . You must also specify the script location using the <code>Linux.ExternalScript.Path</code> property. If the location type is <code>nfs</code> , also use the <code>Linux.ExternalScript.Server</code> property.
L	<code>Linux.ExternalScript.Server</code>	Specifies the name of the NFS server on which the Linux external customization script named in <code>Linux.ExternalScript.Name</code> is located
L	<code>Linux.ExternalScript.Path</code>	Specifies the local path to the Linux customization script or the export path to the Linux customization on the NFS server. The value must begin with a forward slash and not include the file name.

Custom properties table type	Property	Description
L	LoadSoftware	Set to True to make the software jobs listed in Website\Software.txt available to the user requesting the machine.
M	MaximumProvisionedMachines	Specifies the maximum number of linked clones for one machine snapshot. The default is 20.
M	Machine.SSH	Set to True to enable the <b>Connect Using SSH</b> option for Linux machines provisioned from this blueprint. If set to True and the <b>Connect using RDP or SSH</b> machine operation is enabled in the blueprint, all Linux machines that are provisioned from the blueprint display the <b>Connect Using SSH</b> option to entitled users.
M	MaximumProvisionedMachines	Specifies the number of machines that you can provision across all users based on a given blueprint. Most blueprint types are unlimited by default. Linked clone blueprints have a default maximum of 20 machines.
O	Opware.Software.Install	Set to True to allow HP Server Automation to install software
O	Opware.Server.Name	Specifies the fully qualified name of the server automation server.
O	Opware.Server.Username	Specifies the user name provided when a password file in the agent directory was created. This user name requires administrative access to the HP Server Automation instance, for example opwareadmin.
O	Opware.BootImage.Name	Specifies the boot image value as defined in HP Server Automation, for example winpe32 for the 32-bit WinPE image. The property is not required when provisioning by cloning.
O	Opware.Customer.Name	Specifies a customer name value as defined in HP Server Automation, for example MyCompanyName.
O	Opware.Facility.Name	Specifies a facility name value as defined in HP server automation, for example Cambridge
O	Opware.Machine.Password	Specifies the default local administrator password for an operating system sequence WIM image such as Opware.OSSequence.Name as defined in HP Server Automation, for example <b>P@ssword1</b>

Custom properties table type	Property	Description
O	Opware.OSSequence.Name	Specifies the operating system sequence name value as defined in HP Server Automation, for example <b>Windows 2003 WIM</b> .
O	Opware.Realm.Name	Specifies the realm name value as defined in HP Server Automation, for example <b>Production</b> .
O	Opware.Register.Timeout	Specifies the time, in seconds, to wait for creation of a provisioning job to complete.
O	Opware.WOL.Enabled	Set to <b>False</b> for virtual provisioning.
O	Opware.WOL.Delay	Specifies the time, in seconds, to wait before running Wake-On-LAN. This setting is ignored if <code>Opware.WOL.Enabled</code> is set to <b>False</b> .
O	Opware.ProvFail.Notify	(Optional) Specifies the notification e-mail address for HP Server Automation to use in the event of provisioning failure.
O	Opware.ProvFail.Owner	(Optional) Specifies the HP Server Automation user to assign ownership to if provisioning fails, for example <code>opswareadmin</code> .
O	Opware.ProvSuccess.Notify	(Optional) Specifies the notification e-mail address for HP Server Automation to use if provisioning is successful.
O	Opware.ProvSuccess.Owner	(Optional) Specifies the HP Server Automation user to which to assign ownership if provisioning is successful, for example <code>opswareadmin</code> .
P	Plugin.AdMachineCleanup.Execute	Set to <b>True</b> to enable the Active Directory cleanup plug-in. By default, each machine's account is disabled when it is destroyed.
P	Plugin.AdMachineCleanup.UserName	Specifies an Active Directory account user name with sufficient privileges to delete, disable, rename, or move Active Directory accounts. The value must be in <code>domain \username</code> format. This property is required if the vCloud Automation Center manager service does not have these rights in a domain, which can occur when you provision machines in more than one domain.
P	Plugin.AdMachineCleanup.Password	Specifies the password associated to the <code>Plugin.AdMachineCleanup.UserName</code> property. For security, select the <b>Encrypt</b> check box in the user interface.
P	Plugin.AdMachineCleanup.Delete	Set to <b>True</b> to delete the accounts of destroyed machines, instead of disabling them.
P	Plugin.AdMachineCleanup.Domain	Specifies the Active Directory domain name that contains the machine account to be destroyed.

Custom properties table type	Property	Description
P	Plugin.AdMachineCleanup.MoveToOu	Moves the account of destroyed machines to a new Active Directory organizational unit. The value is the organization unit to which you are moving the account. This value must be in ou=OU, dc=dc format
P	Plugin.AdMachineCleanup.RenamePrefix	Renames the accounts of destroyed machines by adding a prefix. The value is the prefix to prepend, for example destroyed_.
P	Pxe.Setup.ScriptName	Specifies a custom EPI PowerShell script to run on the machine before it is booted using the PXE network boot program. The value is the name assigned to the script when it is uploaded to the model manager.
P	Pxe.Clean.ScriptName	Specifies the name of a EPI PowerShell script installed in the vCloud Automation Center model manager, to run on the machine after it is provisioned. The value is the name assigned to the script when it is uploaded to the model manager.
R	RDP.File.Name	Specifies an RDP file from which to obtain settings, for example My_RDP_Settings.rdp. The file must reside in the Website\Rdp subdirectory of the vCloud Automation Center installation directory.
S	Sysprep.Identification.DomainAdmin	Specifies a user name with administrator-level access to the target domain in Active Directory. Do not include the user domain in the credentials that you send to vCloud Director
S	Sysprep.Identification.DomainAdminPassword	Specifies the password to associate with the Sysprep.Identification.DomainAdmin property
S	Sysprep.Identification.JoinDomain	Specifies the name of the domain to join in Active Directory

Custom properties table type	Property	Description
S	SysPrep.Section.Key	<p>Specifies information to be added to the SysPrep answer file on machines during the WinPE stage of provisioning. Information that already exists in the SysPrep answer file is overwritten by these custom properties. Section represents the name of the section of the SysPrep answer file, for example <code>GuiUnattended</code> or <code>UserData</code>. Key represents a key name in the section. For example, to set the time zone of a provisioned machine to West Pacific Standard Time, define the custom property <code>GuiUnattended.UserData.TimeZone</code> and set the value to 275. For a full list of sections, keys, and accepted values, see the System Preparation Utility for Windows documentation. The following <code>Section.Key</code> combinations can be specified for WIM-based provisioning:</p> <ul style="list-style-type: none"> <li>• <code>GuiUnattended</code></li> <li>• <code>AdminPassword</code></li> <li>• <code>EncryptedAdminPassword</code></li> <li>• <code>TimeZone</code></li> <li>• <code>UserData</code></li> <li>• <code>ProductKey</code></li> <li>• <code>FullName</code></li> <li>• <code>ComputerName</code></li> <li>• <code>OrgName</code></li> <li>• <code>Identification</code></li> <li>• <code>DomainAdmin</code></li> <li>• <code>DomainAdminPassword</code></li> <li>• <code>JoinDomain</code></li> <li>• <code>JoinWorkgroup</code></li> </ul>
S	SCCM.Collection.Name	<p>Specifies the name of the SCCM collection that contains the operating system deployment task sequence.</p>

Custom properties table type	Property	Description
S	SCCM.Server.Name	Specifies the fully qualified domain name of the SCCM server on which the collection resides.
S	SCCM.Server.SiteCode	Specifies the site code of the SCCM server.
S	SCCM.Server.UserName	Specifies a user name with administrator-level access to the SCCM server.
S	SCCM.Server.Password	Specifies the password associated with the SCCM.Server.UserName property.
S	SCCM.CustomVariable.Name	Specifies the value of a custom variable, where Name is the name of any custom variable to be made available to the SCCM task sequence after the provisioned machine is registered with the SCCM collection. The value is determined by your choice of custom variable.
S	Snapshot.Policy.AgeLimit	Sets the age limit, in days, for snapshots that can be applied to machines.
S	Snapshot.Policy.Limit	Sets the depth limit of snapshots that can be created for machines.
V	VirtualMachine.Admin.UseGuestAgent	If the guest agent is installed as a service on a template for cloning, set to True on the machine blueprint to enable the guest agent service on machines cloned from that template. Set to False to disable the guest agent.
V	VirtualMachine.Admin.NameCompletion	Specifies the domain name to include in the fully qualified domain name of the machine that the RDP files generate for the Connect Using RDP option. For example, set the value to myCompany.com to generate the FDQN my-machine-name.myCompany.com in the RDP file.
V	VirtualMachine.Admin.ConnectionAddress	Specifies the RDP connection address of the machine to which an RDP file will be downloaded when the <b>Connect Using RDP</b> option is used or attached to automatic e-mails. Do not use in a blueprint or build profile unless <b>Prompt User</b> is enabled and no default value is supplied
V	VirtualMachine.Admin.ThinProvision	Determines whether thin provisioning is used on ESX compute resources using local or iSCSI storage. Set to True to use thin provisioning. Set to False to use standard provisioning. This property is for virtual provisioning.
V	VirtualMachine.Admin.CustomizeGuestOSDelay	Specifies the time to wait after customization is complete and before starting the guest operating system customization. The value must be in HH:MM:SS format.

Custom properties table type	Property	Description
V	VirtualMachine.Admin.NetworkInterfaceType	<p>Indicates the network interface type. Use to modify the network settings of a newly provisioned virtual machine. The following options are available:</p> <ul style="list-style-type: none"> <li>• E1000 (default)</li> <li>• VirtIO</li> <li>• RTL8139</li> <li>• RTL8139 VirtIO</li> </ul> <p>This property is for virtual provisioning.</p>
V	VirtualMachine.Admin.OverrideVncConsole	<p>Specifies how the consoles of virtual machines are accessed. The default display type is VNC. Set to <code>True</code> to use the KVM (RHEV) display type, Spice. This property is available for KVM (RHEV) blueprints</p>
V	VirtualMachine.Admin.Name	<p>Specifies the generated machine name. You can use this property can be used when creating custom workflows or plug-ins. The value in the blueprint or build profile has no effect on this property.</p>
V	VirtualMachine.Admin.UUID	<p>Specifies the UUID of the machine. The value is recorded by the guest agent when the machine is created, then it becomes read-only. The value in the blueprint or build profile has no effect on this property.</p>
V	VirtualMachine.Admin.AgentID	<p>Specifies the UUID of the guest agent. The value is recorded by the guest agent when the machine is created, then it becomes read-only. The value in the blueprint or build profile has no effect on this property.</p>
V	VirtualMachine.Admin.Owner	<p>Specifies the user name of the machine owner.</p>
V	VirtualMachine.Admin.Approver	<p>Specifies the user name of the machine owner</p>
V	VirtualMachine.Admin.Description	<p>Specifies the user name of the group manager who approved the machine request.</p>
V	VirtualMachine.Admin.AdministratorEmail	<p>Specifies the description of the machine as entered or modified by its owner or an administrator</p>
V	VirtualMachine.Admin.AdministratorEmail	<p>Specifies the manager e-mail addresses or Active Directory accounts for the business group of the provisioning blueprint. Multiple e-mail addresses are separated by a comma, for example <code>JoeAdmin@VMware.com,WeiMgr@VMware.com</code>.</p>



Custom properties table type	Property	Description
V	<code>VirtualMachine.Host.TpmEnabled</code>	Limits virtual machine placement to hosts that have a Trust Protection Module (TPM) device installed and recognized by ESX and vSphere. The default value is <code>False</code> . All hosts in a cluster must have a <b>Trust Protection Module (TPM)</b> device installed. If no acceptable hosts or clusters are found, the machine cannot be provisioned until this property is removed.
V	<code>VirtualMachine.Admin.Hostname</code>	Specifies the name of the compute resource on which the machine resides.
V	<code>VirtualMachine.Admin.ClusterName</code>	Specifies the name of the cluster that contains the compute resource on which the machine resides.
V	<code>VirtualMachine.Admin.ApplicationID</code>	List the application IDs that can be assigned to a machine.
V	<code>VirtualMachine.Admin.AddOwnerToAdmins</code>	Set to <code>True</code> to add the machine's owner, as specified by the <code>VirtualMachine.Admin.Owner</code> property, to the local administrators group on the machine. This property is not available for provisioning by cloning.
V	<code>VirtualMachine.Admin.AllowLogin</code>	Set to <code>True</code> (default) to add to the local remote desktop users group the machine's owner, as specified by the <code>VirtualMachine.Admin.Owner</code> property.
V	<code>VirtualMachine.Agent.CopyToDisk</code>	Set to <code>True</code> (default) to copy the guest agent executable file to <code>%SystemDrive%\VRM\Build\Bin</code> on the machine's disk.
V	<code>VirtualMachine.Agent.GuiRunOnce</code>	Set to <code>False</code> for the Linux agent to stop the provisioning workflow. This property is for kickstart provisioning with Linux or AutoYaST.
V	<code>VirtualMachine.Agent.Reboot</code>	Set to <code>True</code> (default) to specify that the guest agent restarts the machine following installation of the guest operating system.
V	<code>VirtualMachine.CDROM.Attach</code>	Set to <code>False</code> to provision the machine without a CD-ROM device. The default is <code>True</code> .
V	<code>VirtualMachine.Customize.WaitComplete</code>	Set to <code>True</code> to hold the provisioning workflow until customizations are finished.
V	<code>VirtualMachine.Admin.DiskInterfaceType</code>	Indicates the type of disk drivers. The following disk drivers are supported: <ul style="list-style-type: none"> <li>• IDE (default)</li> <li>• VirtIO</li> </ul> This property is for virtual provisioning

Custom properties table type	Property	Description
V	<code>VirtualMachine.DiskN.Letter</code>	Specifies the letter or mount point of a machine's disk N. The default is C. For example, to specify the letter D for Disk 1, define the custom property as <code>VirtualMachine.Disk1.Letter</code> and enter the value D. Disk numbering must be sequential.
V	<code>VirtualMachine.DiskN.Size</code>	Defines the size in GB of disk N. For example, to give a size of 150 GB to a disk G, define the custom property <code>VirtualMachine.Disk0.Size</code> and enter a value of 150. Disk numbering must be sequential. By default a machine has one disk referred to by <code>VirtualMachine.Disk0.Size</code> , where size is specified by the Storage option on the Build Information tab of the V blueprint from which the machine is provisioned. The value in the Storage option on the Build Information tab overwrites the value in the <code>VirtualMachine.Disk0.Size</code> property. The <code>VirtualMachine.Disk0.Size</code> property is not available as a custom property because of its relationship with the Storage option on the Build Information tab. More disks can be added by specifying <code>VirtualMachine.Disk1.Size</code> , <code>VirtualMachine.Disk2.Size</code> and so on. <code>VirtualMachine.Admin.TotalDiskUsage</code> always represents the total of the <code>DiskN.Size</code> properties plus the <code>VMware.Memory.Reservation</code> size allocation.
V	<code>VirtualMachine.DiskN.Label</code>	Specifies the label of a machine's disk. The disk label maximum is 32 characters. Disk numbering must be sequential
V	<code>VirtualMachine.DiskN.Active</code>	Set to <code>True</code> (default) to specify that the machine's disk N is active. Set to <code>False</code> to specify that the machine's disk N is not active.
V	<code>VirtualMachine.DiskN.FS</code>	Specifies the file system of the machine's disk N. The options are NTFS (default), FAT and FAT32.
V	<code>VirtualMachine.DiskN.Percent</code>	Specifies the percentage of the disk N to be formatted by a guest agent for the machine's use. That machine cannot use the remaining portion of the disk.

Custom properties table type	Property	Description
V	<code>VirtualMachine.DiskN.StorageReservationPolicy</code>	Specifies a storage reservation policy to find storage for disk N. Also assigns a storage reservation policy to a volume. You can use this disk property only with linked clone blueprints. To use this property, substitute the volume number for N in the property name and specify a storage reservation policy as the value. This property is equivalent to the Storage Reservation Policy drop-down menu in the Volumes list on the Build Information tab of the blueprint page. Disk numbering must be sequential.
V	<code>VirtualMachine.DiskN.StorageReservationPolicyMode</code>	Allocates disk N on the best available storage reservation policy.
V	<code>VirtualMachine.Rdp.File</code>	Specifies the RDP file that contains settings to be used when opening an RDP link to the machine. Can be used together with, or as an alternative to, <code>VirtualMachine.Rdp.SettingN</code> . The file must be located in <code>vcac_installation_dir\Website\Rdp</code> where <code>vcac_install_dir</code> is the server install directory, for example <code>%SystemDrive%\Program Files x86\VMware\vcac\Server\Rdp\console.rdp</code> . You must create the Rdp directory
V	<code>VirtualMachine.CPU.Count</code>	Specifies the number of CPUs allocated to a machine. The default is the value specified by the # CPUs setting on the blueprint's Build Information tab.
V	<code>VirtualMachine.Memory.Size</code>	Specifies the size of the machine's memory in MB. The default is the value specified by the Memory option on the blueprint's Build Information tab.
V	<code>VirtualMachine.Admin.TotalDiskUsage</code>	Specifies the total disk space that the machine uses, including all disks as specified by the <code>VirtualMachine.DiskN.Size</code> properties and the swap file as specified by the <code>VMware.Memory.Reservation</code> property.
V	<code>VirtualMachine.Storage.Name</code>	Identifies the storage path on which the machine resides. The default is the value specified in the reservation that was used to provision the machine.
V	<code>VirtualMachine.DiskN.Storage</code>	Specifies that datastore on which to place the machine disk N. This property is also used to add a single datastore to a linked clone blueprint. N is the index (starting at 0) of the volume to assign. Type the name of the datastore to assign to the volume in the Value text box. This is the datastore name as it appears in the Storage Path column on the Edit Compute Resource page. Disk numbering must be sequential.

Custom properties table type	Property	Description
V	<code>VirtualMachine.DiskN.VMwareType</code>	<p>Specifies the VMware disk mode of the machine's disk N. The following options are available:</p> <ul style="list-style-type: none"> <li>• persistent</li> <li>• independent_persistent</li> <li>• independent_nonpersistent</li> </ul> <p>For details, see <code>VirtualDeviceDeviceBackingOption</code> data object help.</p>
V	<code>VirtualMachine.EPI.Type</code>	<p>Specifies the type of external provisioning infrastructure. Set to <b>BMC</b> for BMC BladeLogic integration. Set to <b>CitrixProvisioning</b> for Citrix provisioning server integration</p>
V	<code>VirtualMachine.NetworkN.Address</code>	<p>Specifies the IP address of a network device N in a machine provisioned with a static IP address.</p>
V	<code>VirtualMachine.NetworkN.MacAddressType</code>	<p>Indicates whether the MAC address of network device N is auto-generated or user-defined. This property is available for cloning. The default value is generated. If the value is static, you must also use <code>VirtualMachine.NetworkN.MacAddress</code> to specify the MAC address.</p>
V	<code>VirtualMachine.NetworkN.MacAddress</code>	<p>Specifies the MAC address of a network device N. This property is available for cloning. If the value of <code>VirtualMachine.NetworkN.MacAddressType</code> is generated, this property contains the generated address. If the value of <code>VirtualMachine.Network.N.MacAddressType</code> is static, this property specifies the MAC address. For virtual machines provisioned on ESX server hosts, the address must be in the range specified by VMware. For details, see vSphere documentation.</p>
V	<code>VirtualMachine.NetworkN.Name</code>	<p>Specifies the network to which a network device N in a virtual machine is attached. By default, a network is assigned from the network paths available on the reservation on which the machine is provisioned. You can ensure that a network device is connected to a specific network by setting the value of this property to the name of a network on an available reservation.</p>
V	<code>VirtualMachine.NetworkN.PortID</code>	<p>Specifies the port ID to use for network device N when using a dvPort group with a vSphere distributed switch.</p>

Custom properties table type	Property	Description
V	VirtualMachine.NetworkN.ProfileName	<p>Specifies the name of a network profile from which to assign a static IP address to network device Nor from which to obtain the range of static IP addresses that can be assigned to network device N of a cloned machine, where N=0 for the first device, 1 for the second, and so on. If a network profile is specified in the network path in the reservation on which the machine is provisioned, a static IP address is assigned from that network profile. You can ensure that a static IP address is assigned from a specific profile by setting the value of this property to the name of a network profile. With WIM-based provisioning for virtual machines, you can use this property to specify a network profile and network interface or you can use the Network section of the Virtual Reservation page. You can also assign the network interface to a virtual network using the VirtualMachine.NetworkN.Name custom property. With WIM-based provisioning for physical machines, you must specify the network profile with the VirtualMachine.NetworkN.ProfileName custom property. You must also specify a value with the VirtualMachine.NetworkN.Name custom property, although vCloud Automation Center does not use the value. The following attributes of the network profile are available to enable static IP assignment in a cloning blueprint:</p> <ul style="list-style-type: none"> <li>• VirtualMachine.NetworkN.SubnetMask</li> <li>• VirtualMachine.NetworkN.Gateway</li> <li>• VirtualMachine.NetworkN.PrimaryDns</li> <li>• VirtualMachine.NetworkN.SecondaryDns</li> <li>• VirtualMachine.NetworkN.PrimaryWins</li> <li>• VirtualMachine.NetworkN.SecondaryWins</li> <li>• VirtualMachine.NetworkN.DnsSuffix</li> <li>• VirtualMachine.NetworkN.DnsSearchSuffixes</li> </ul>

Custom properties table type	Property	Description
V	VirtualMachine.Rdp.SettingN	Configures specific RDP settings. N is a unique number used to distinguish one RDP setting from another. For example, to specify the Authentication Level so that no authentication requirement is specified, define the custom property <code>VirtualMachine.Rdp.Setting1</code> and set the value <code>authentication level:i:3</code> . For a list of available settings and correct syntax, see the Microsoft Windows RDP documentation.
V	VirtualMachine.Software0.ScriptPath	Specifies the path to an application's install script. The path must be a valid absolute path as seen by the guest operating system and must include the name of the <code>.bat</code> file. You can pass custom property values as parameters to the script by inserting <code>{CustomPropertyName}</code> in the path string. For example, if you have a custom property named <code>ActivationKey</code> whose value is <code>1234</code> , the script path is <code>D:\InstallApp.bat -key {ActivationKey}</code> . The guest agent runs the command <code>D:\InstallApp.bat -key 1234</code> . Your script file can then be programmed to accept and use this value.
V	VbScript.PreProvisioning.Name	Specifies the full path of a Visual Basic script to be run before a machine is provisioned. For example, <code>%System-Drive%\Program Files (x86)\VMware\VCAC_Agents\EPI_Agent\Scripts\SendEmail.vbs</code> . The script file must reside on the system on which the Visual Basic script EPI agent is installed.
V	VbScript.PostProvisioning.Name	Specifies the full path of a Visual Basic script to be run after a machine is provisioned. For example, <code>%System-Drive%\Program Files (x86)\VMware\VCAC_Agents\EPI_Agent\Scripts\SendEmail.vbs</code> . The script file must reside on the system on which the Visual Basic script EPI agent is installed.
V	VbScript.UnProvisioning.Name	Specifies the full path of a Visual Basic script to be run when a machine is destroyed. For example, <code>%SystemDrive%\Program Files (x86)\VMware\VCAC_Agents\EPI_Agent\Scripts\SendEmail.vb</code> . The script file must reside on the system on which the Visual Basic script EPI agent is installed.

Custom properties table type	Property	Description
V	VCloud.Template.MakeIdenticalCopy	<p>Set to True to clone an identical copy of the vApp template in vCloud Director and provision the results in vCloud Automation Center. This ignores all settings specified in the blueprints except the name of the vApp and its virtual machines. The storage path specified in the vApp template during cloning is used, even if a different storage path is specified in a vApp component blueprint or when requesting a vApp machine. Set to False to clone a copy of the vApp template with settings specified by the vApp and vApp component blueprints. The storage path specified in a vApp component blueprint, operating system or when requesting a vApp machine, is used if the Make Identical Copy option in the vApp template properties was selected.</p>
V	VCNS.SecurityGroup.Names.name	<p>Specifies the vCloud Networking and Security security group or groups to which the virtual machine is assigned during provisioning. The value is a security group name or a list of names separated by commas. Names are case-sensitive. Appending a name allows you to create multiple versions of the property, which can be used separately or in combination. For example, the following properties might list security groups intended for general use, for the sales force, and for support:</p> <ul style="list-style-type: none"> <li>• VCNS.SecurityGroup.Names</li> <li>• VCNS.SecurityGroup.Names.sales</li> <li>• VCNS.SecurityGroup.Names.spport</li> </ul>

Custom properties table type	Property	Description
V	VCNS.LoadBalancerEdgePool.Names.name	<p>Specifies the vCloud Networking and Security load balancing pools to which the virtual machine is assigned during provisioning. The virtual machine is assigned to all service ports of all specified pools. The value is an edge/pool name or a list of edge/pool names separated by commas. Names are case-sensitive. Appending a name allows you to create multiple versions of a custom property. For example, the following properties might list load balancing pools set up for general use and machines with high, moderate, and low performance requirements:</p> <ul style="list-style-type: none"> <li>• VCNS.LoadBalancerEdgePool.Names</li> <li>• VCNS.LoadBalancerEdgePool.Names.moderate</li> <li>• VCNS.LoadBalancerEdgePool.Names.high</li> <li>• VCNS.LoadBalancerEdgePool.Names.low</li> </ul>
V	VMware.VirtualCenter.OperatingSystem	<p>Specifies the vCenter Server guest operating system version (<code>VirtualMachineGuestOsIdentifier</code>) with which vCenter Server creates the machine. This operating system version must match the operating system version to be installed on the provisioned machine. Administrators can create build profiles using one of several property sets, for example, <code>VMware[OS_Version]Properties</code>, that are predefined to include the correct <code>VMware.VirtualCenter.OperatingSystem</code> values. This property is for virtual provisioning. When this property has a non-Windows value, the Connect Using RDP option is disabled. The property can be used in a virtual, cloud or physical blueprint. For a list of currently accepted values, see the VMware® vCenter Server™ documentation.</p>



Custom properties table type	Property	Description
V	VMware.SCSI.Type	<p>For vApps, specifies the SCSI machine type using one of the following case-sensitive values:</p> <ul style="list-style-type: none"> <li>• <b>buslogic</b>: Use BusLogic emulation for the virtual disk</li> <li>• <b>lsilogic</b>: Use LSILogic emulation for the virtual disk (default).</li> <li>• <b>lsilogicsas</b>: Use LSILogic SAS 1068 emulation for the virtual disk.</li> <li>• <b>VirtualSCSI</b>: Use para-virtualization emulation for the virtual disk.</li> <li>• <b>none</b>: Use if a SCSI controller does not exist for this machine</li> </ul>
	VMware.SCSI.Type	<p>For vSphere only, specifies the SCSI machine type using one of the following case-sensitive values:</p> <ul style="list-style-type: none"> <li>• <b>buslogic</b>: Use BusLogic emulation for the virtual disk</li> <li>• <b>lsilogic</b>: Use LSILogic emulation for the virtual disk (default).</li> <li>• <b>lsilogicsas</b>: Use LSILogic SAS 1068 emulation for the virtual disk.</li> <li>• <b>VirtualSCSI</b>: Use para-virtualization emulation for the virtual disk.</li> <li>• <b>none</b>: Use if a SCSI controller does not exist for this machine</li> </ul>
V	VMware.SCSI.Sharing	Specifies the sharing mode of the machine's VMware SCSI bus. Possible values are based on the VirtualSCSISharing ENUM value.
V	VMware.Memory.Reservation	Specifies the size of the machine's swap file.
V	VMware.VirtualCenter.Folder	Name of the inventory folder in the datacenter in which the machine is created. The default is VRM. Can be a path with multiple folders, for example production\email servers\. A proxy agent creates the folder in vCenter Server if it does not exist. VMware and vSphere only
V	VRM.Software.IdNNNN	Specifies a software job or policy to be applied to all machines provisioned from the blueprint. Set the value to job_type=job_path, where job_type is the numeral that represents the BMC BladeLogic job type and job_path is the location of the job in BMC BladeLogic. For example: 4=/Utility/putty. NNNN is a number from 1000 to 1999.

Custom properties table type	Property	Description
V	VRM.Software.IdNNNN	Optionally specify an HP Server Automation policy to be applied to all machines provisioned from the blueprint. NNNN is a number from 1000 to 1999.
V	VRM.DataCenter.Policy	Specifies whether provisioning must use a compute resource associated with a particular location, or if any location is suitable. To enable this feature you must add data center to a location file. Associate each compute resource with a location. Set to Exact to provision a requested machine on a compute resource associated with the location specified on the blueprint. If a compute resource with sufficient capacity and associated with that location is not available, then provisioning fails. Set to NonExact (default) to provision a requested machine on a compute resource with sufficient capacity and associated with the location specified on the blueprint. If that compute resource is not available, then use the next available compute resource with sufficient capacity without regard to location.
V	VMware.Network.Type	<p>Specifies a network adapter type. The following adapter type values are available:</p> <ul style="list-style-type: none"> <li>• <b>Flexible</b> (default)</li> <li>• <b>VirtualPcNet32</b>. This type is not compatible with vSphere</li> <li>• <b>E1000</b> or <b>VirtualE1000</b></li> <li>• <b>VMXNET</b> or <b>VirtualVMXNET</b></li> <li>• <b>VMXNET2</b></li> <li>• <b>VMXNET3</b></li> </ul> <p>Set to E1000 when provisioning Windows 32-bit virtual machines on ESX server hosts to ensure that machines are created with the correct network adapter. This property is not used in physical provisioning.</p>
V	VirtualMachine.VDI.Type	Specifies the type of virtual desktop infrastructure. For XenDesktop provisioning, set to XenDesktop.

Custom properties table type	Property	Description
V	VDI.Server.Website	Specifies the server name of the Citrix Web interface site to use in connecting to the machine. If the value of VDI.Server.Name is a XenDesktop farm, this property must have an appropriate value or the machine owner cannot connect to the machine using XenDesktop. If this property is not specified, the VDI.Server.Name property determines the desktop delivery controller to connect to, which must be the name of a server that hosts a desktop delivery controller.
V	VDI.Server.Name	Specifies the server name, which hosts the desktop delivery controller, to register with, or the name of a XenDesktop farm that contains desktop delivery controllers with which to register. If the value is a farm name, the VDI.Server.Website property value must be the URL of an appropriate Citrix web interface site to use in connecting to the machine. If the value is a server name, and at least one general XenDesktop VDI agent was installed without specifying a desktop delivery controller server, this value directs the request to the desired server. If the value is a server name, and only dedicated XenDesktop VDI agents for specific DDC servers were installed, this value must exactly match the server name configured for a dedicated agent
V	VDI.Server.Group	For XenDesktop 5, specifies the name of the XenDesktop group to add machines to and the name of the catalog to which the group belongs, in the group_name;catalog_name format. For XenDesktop 4, specifies the name of the XenDesktop group to which machines are to be added. XenDesktop 4 preassigned groups are supported.
V	VDI.ActiveDirectory.Interval	Specifies an optional interval value in time span format for virtual desktop infrastructure machine Active Directory registration check. The default value is 00:00:15 (15 seconds).
V	VDI.ActiveDirectory.Interval	Specifies an optional timeout value in time span format for virtual desktop infrastructure machine Active Directory registration check. The default value is 00:30:00 (30 minutes.)
V	VDI.ActiveDirectory.Delay	Specifies an optional delay time value in time span format between successfully adding a machine to Active Directory and initiation of XenDesktop registration. The default value is 00:00:05 (5 seconds).

Custom properties table type	Property	Description
Z	Xen.Platform.Viridian	For virtual provisioning, set to False when you provision Windows virtual machines on a XenServer host or pool. The default is True. The property is not used in physical provisioning.

## Other sources

The following are links to sites dedicated to vRealize Automation where you can find detailed information on specific subjects:

- <http://dailyhypervisor.com/>
- <http://www.vmtocloud.com/>
- <http://cloudyautomation.com/>
- <http://cloudrelevant.com/>
- <http://virtumaster.com/>
- <http://www.vcoteam.info/>
- <http://dailyhypervisor.com/vcloud-automation-center-vcac-5-2-custom-hostnaming-extension/>
- Microsoft Deployment Toolkit at <http://www.microsoft.com/en-us/download/details.aspx?id=40796>
- Microsoft System Center 2012 R2 Configuration Manager at <http://www.microsoft.com/en-us/server-cloud/products/system-center-2012-r2-configuration-manager/>
- REST API tutorial at <http://www.restapitutorial.com>



# Index

## A

- Active Directory (AD)** 11
- Advanced Service Designer (ASD)** 6
- AppD Bootstrap agent**
  - installing, on VM 123-126
- AppD web portal**
  - configuring 121-123
- Application Director (AppD)** 115
- application programming interface (API)** 132
- Approval Policies**
  - about 109-113
  - Post Approval Policy, creating 114

## B

- Blueprint**
  - creating, with MDT 93-95
- Blueprint options**
  - Display Location on Request 82
  - Master 82
  - Shared Blueprint 82
- Business group** 33

## C

- catalog management**
  - about 37
  - actions 37
  - catalog items 37
  - entitlements 37
  - services 37
- Certificate Authority** 28
- CitrixProvisioning** 223
- clone Blueprint**
  - creating 88-90

## CloudClient

- about 16
- features 16
- compute resources** 33
- core components, vRealize Automation**
  - about 10, 11
  - IaaS server 11
  - vRealize Automation Appliance 11
  - vRealize Automation Identity Appliance 11
- customized workflow**
  - CPU and memory hot add workflow, creating 165-172
  - testing 164, 165
- Custom Properties**
  - about 36
  - reference 36
- custom workflow**
  - creating, with vRealize orchestrator client 149-164

## D

- Database as a Service (DBaaS)** 1
- DEM components, IaaS server**
  - about 12
  - Orchestrator DEM 12
  - Worker DEM 12
- deployed service requests**
  - modifying 194, 195
- design elements, pilot**
  - about 28
  - certificates 28
  - concurrent provisioning 28
  - deployment size 28
  - firewall 28
  - hardware sizing 28

- IT Business Management Suite
  - Standard Edition 28
- network 28
- portal layout 28
- scale 28
- storage 28

## E

- editions, vRealize Automation 6.2**
  - features 4, 5
- end user web portal**
  - customizing 195
- ESXi software 19**
- Everything as a Service (XaaS) 1**
- example customized workflow**
  - defining 140
- example design exercise**
  - about 38
  - blueprints 43
  - catalog items 43
  - compute resources, calculating 41
  - network considerations 42
  - provisioned machines, naming 42
  - requisites 39
  - services 43
  - storage amount, calculating 41
  - vRealize Automation components 40

## F

- Fabric groups 34**
- features, vRealize Automation 6.2**
  - about 14, 17
  - blueprints 18
  - build profiles 18
  - clone deployment 18
  - CloudClient 16
  - costs, associating to resources 20
  - Infrastructure Organizer 19
  - integration with vROps 16
  - Kickstart deployment 19
  - logs, for auditing and compliance 20
  - pay-as-you-go support 17
  - SCCM deployment 19
  - SSO authentication 22
  - Tenants 21

- VMware Horizon View desktops
  - integration 21
- vRealize Orchestration (vRO) 21
- vROps integration 15
- WIM deployment 18
- Fully Qualified Domain Name (FQDN) 118**

## H

- Hands-on Labs (HOL) 10**
- high-level architecture, vRealize Automation 6.2 13**
- HTTP methods**
  - using, for implementing RESTful API 132

## I

- IaaS server**
  - about 11
  - agents 12
  - configuring 58-62
  - DEM components 12
  - downloading 55
  - installing 55, 58
  - model manager 12
- Infrastructure as a Service (IaaS) 1**
- Infrastructure Organizer 19**
- Internet Small Computer System Interface (iSCSI) 28**
- IT as a Service (ITaaS) 1**

## K

- Kickstart 6, 99**
- Kickstart Blueprint**
  - creating 100-103
- KiTTY 194**

## L

- Lightweight Directory Access Protocol (LDAP) 11**
- Linked Clone 122**
- linked clone Blueprint**
  - creating 80-84
  - preparing, for provisioning 84-86
  - provisioning 86, 87

## M

**machine prefixes** 42

**Microsoft Deployment Toolkit (MDT)**  
server 91

**Microsoft Distributed Transaction  
Coordinator (MSDTC)** 11

**model manager, IaaS server** 12

**monitoring**

about 204

Audit Log 204

Distributed Execution Status 204

Log 204

Workflow History 204

**multi-machine Blueprint**

about 103

creating 103, 104

## N

**Namespace Extension (NSX)** 14

**NAT (Network Address Translation)** 35

**Network Attached Storage (NAS)** 28

**network profile** 35

## O

**Onyx**

about 174

reference 174

## P

**pay-as-you-go support**

features 17

**Payment Card Industry (PCI)** 3

**PE Builder Utility**

reference 91

**physical machines**

provisioning 104

**pilot**

about 27

deployment 27

design elements 28

process 29

**plugins**

installing, to access VMs through web  
portal 193, 194

**Proof Of Concept (POC)** 26

**Proof of Technology (POT)** 27

**Property Dictionary settings**

configuring 104-106

**provisioned machine**

destroying 106

## R

**Red Hat Enterprise Linux (RHEL)** 3

**Representational State Transfer.** *See* REST

**reservation policy** 35

**reservations**

about 34, 35

network profile 35

reservation policy 35

**REST**

about 130

cacheable 131

client-server 131

Code on Demand 132

layered system 132

stateless 131

uniform interface 130

**RESTful** 130

## S

**Sarbanes-Oxley (SOX)** 3

**SCCM Blueprint**

creating 96-99

**SCCM (System Center Configuration  
Manager)** 96

**service requests**

creating 190-192

monitoring 192

**Single Sign-On (SSO) authentication** 11

**SQL**

configuring, for provisioning 126-128

**Storage Area Network (SAN)** 28



**T**

- Tenants 21
- Trust Protection Module (TPM) 220

**U**

- UAC settings (User Access Control) 59
- uniform interface
  - guiding principles 130
- URIs (Uniform Resource Identifiers) 130
- User Account Control (UAC) 125

**V**

- vCenter Operations (vCops) 4
- vCenter Orchestrator (vCO) 6
- vCloud Automation Center custom properties
  - AD.Lookup.Department 209
  - Amazon.ElasticLoadBalancer.Names 209
  - Amazon.Instance.Id 209
  - BMC.AddServer.Delay 209
  - BMC.AddServer.Retry 209
  - BMC.Service.Profile 209
  - BMC.Software.BatchLocation 209
  - BMC.Software.Install 209
  - Cisco.Organization.DN 209
  - CloneFrom 209
  - CloneSpec 210
  - Command.DiskPart.Options 210
  - Command.FormatDisk.Options 210
  - EPI.Server.Collection 211
  - EPI.Server.Name 210
  - EPI.Server.Port 211
  - EPI.Server.Site 211
  - EPI.Server.Store 211
  - EPI.Server.VDiskName 211
  - Hostname 211
  - Hyperv.Network.Type 211
  - Image.ISO.Location 211
  - Image.ISO.Name 212
  - Image.ISO.Password 212
  - Image.ISO.UserName 212
  - Image.Network.Letter 213
  - Image.Network.Password 212
  - Image.Network.User 212
  - Image.WIM.Index 212
  - Image.WIM.Name 212
  - Image.WIM.Path 212
  - Infrastructure.Admin.ADPassword 213
  - Infrastructure.Admin.ADUUser 213
  - Infrastructure.Admin.DefaultDomain 213
  - Infrastructure.Admin.Machine
    - ObjectOU 213
  - Infrastructure.ResourcePool.Name 213
  - Linux.ExternalScript.LocationType 213
  - Linux.ExternalScript.Name 213
  - Linux.ExternalScript.Path 213
  - Linux.ExternalScript.Server 213
  - LoadSoftware 214
  - Machine.SSH 214
  - MaximumProvisionedMachines 214
  - Opsware.BootImage.Name 214
  - Opsware.Customer.Name 214
  - Opsware.Facility.Name 214
  - Opsware.Machine.Password 214
  - Opsware.OSSequence.Name 215
  - Opsware.ProvFail.Notify 215
  - Opsware.ProvFail.Owner 215
  - Opsware.ProvSuccess.Notify 215
  - Opsware.ProvSuccess.Owner 215
  - Opsware.Realm.Name 215
  - Opsware.Register.Timeout 215
  - Opsware.Server.Name 214
  - Opsware.Server.Username 214
  - Opsware.Software.Install 214
  - Opsware.WOL.Delay 215
  - Opsware.WOL.Enabled 215
  - Plugin.AdMachineCleanup.Delete 215
  - Plugin.AdMachineCleanup.Domain 215
  - Plugin.AdMachineCleanup.Execute 215
  - Plugin.AdMachineCleanup.MoveToOu 216
  - Plugin.AdMachineCleanup.Password 215
  - Plugin.AdMachineCleanup.Rename
    - Prefix 216
  - Plugin.AdMachineCleanup.UserName 215
  - Pxe.Clean.ScriptName 216
  - Pxe.Setup.ScriptName 216
  - RDP.File.Name 216
  - SCCM.Collection.Name 217
  - SCCM.CustomVariable.Name 218
  - SCCM.Server.Name 218

SCCM.Server.Password 218  
 SCCM.Server.SiteCode 218  
 SCCM.Server.UserName 218  
 Snapshot.Policy.AgeLimit 218  
 Snapshot.Policy.Limit 218  
 Sysprep.Identification.DomainAdmin 216  
 Sysprep.Identification.DomainAdmin  
     Password 216  
 Sysprep.Identification.JoinDomain 216  
 SysPrep.Section.Key 217  
 VbScript.PostProvisioning.Name 225  
 VbScript.PreProvisioning.Name 225  
 VbScript.UnProvisioning.Name 225  
 VCloud.Template.MakelIdenticalCopy 226  
 VCNS.LoadBalancerEdge  
     Pool.Names.name 227  
 VCNS.SecurityGroup.Names.name 226  
 VDI.ActiveDirectory.Delay 230  
 VDI.ActiveDirectory.Interval 230  
 VDI.Server.Group 230  
 VDI.Server.Name 230  
 VDI.Server.Website 230  
 VirtualMachine.Admin.AddOwner  
     ToAdmins 220  
 VirtualMachine.Admin.  
     AdministratorEmail 219  
 VirtualMachine.Admin.AgentID 219  
 VirtualMachine.Admin.AllowLogin 220  
 VirtualMachine.Admin.ApplicationID 220  
 VirtualMachine.Admin.Approver 219  
 VirtualMachine.Admin.ClusterName 220  
 VirtualMachine.Admin.ConnectionAddress  
     218  
 VirtualMachine.Admin.Customize  
     GuestOSDelay 218  
 VirtualMachine.Admin.Description 219  
 VirtualMachine.Admin.DiskInterface  
     Type 220  
 VirtualMachine.Admin.Hostname 220  
 VirtualMachine.Admin.Name 219  
 VirtualMachine.Admin.Name  
     Completion 218  
 VirtualMachine.Admin.Network  
     InterfaceType 219  
 VirtualMachine.Admin.OverrideVnc  
     Console 219  
 VirtualMachine.Admin.Owner 219  
 VirtualMachine.Admin.ThinProvision 218  
 VirtualMachine.Admin.TotalDisk  
     Usage 222  
 VirtualMachine.Admin.UseGuest  
     Agent 218  
 VirtualMachine.Admin.UUID 219  
 VirtualMachine.Agent.CopyToDisk 220  
 VirtualMachine.Agent.GuiRunOnce 220  
 VirtualMachine.Agent.Reboot 220  
 VirtualMachine.CDROM.Attach 220  
 VirtualMachine.CPU.Count 222  
 VirtualMachine.Customize.Wait  
     Complete 220  
 VirtualMachine.DiskN.Active 221  
 VirtualMachine.DiskN.FS 221  
 VirtualMachine.DiskN.Label 221  
 VirtualMachine.DiskN.Letter 221  
 VirtualMachine.DiskN.Percent 221  
 VirtualMachine.DiskN.Size 221  
 VirtualMachine.DiskN.Storage 222  
 VirtualMachine.DiskN.Storage  
     ReservationPolicy 222  
 VirtualMachine.DiskN.Storage  
     ReservationPolicyMode 222  
 VirtualMachine.DiskN.VMwareType 223  
 VirtualMachine.EPI.Type 223  
 VirtualMachine.Host.TpmEnabled 220  
 VirtualMachine.Memory.Size 222  
 VirtualMachine.NetworkN.Address 223  
 VirtualMachine.NetworkN.Mac  
     Address 223  
 VirtualMachine.NetworkN.Mac  
     AddressType 223  
 VirtualMachine.NetworkN.Name 223  
 VirtualMachine.NetworkN.PortID 223  
 VirtualMachine.NetworkN.Profile  
     Name 224  
 VirtualMachine.Rdp.File 222  
 VirtualMachine.Rdp.SettingN 225  
 VirtualMachine.Software0.ScriptPath 225  
 VirtualMachine.Storage.Name 222  
 VirtualMachine.VDI.Type 229  
 VMware.Memory.Reservation 228  
 VMware.Network.Type 229  
 VMware.SCSI.Sharing 228

- VMware.SCSI.Type 228
- VMware.VirtualCenter.Folder 228
- VMware.VirtualCenter.Operating System 227
- VRM.DataCenter.Policy 229
- VRM.Software.IdNNNN 228, 229
- Xen.Platform.Viridian 231
- vCloud Automation Center (vCAC) 3**
- Virtual Appliance (vApp) 6**
- Virtual Desktop Infrastructure (VDI) 12**
- Virtual Machine Remote Control (VMRC) plugin 199**
- virtual machines (VMs) 12**
- VMware documentation 208**
- VMware Knowledge Base (KB) 199**
- VMware Remote Control (VMRC) plugin 193**
- vRA environment**
  - references 207
- vRA Guest Agent**
  - URL 102
- vRealize Automation (vRA)**
  - about 1
  - benefits 2, 3
  - configuring 64-66
  - core components 10, 11
  - edition, selecting 10
  - infrastructure, configuring 66-77
  - references 231
  - upgrading 63, 64
- vRealize Automation 6.2**
  - about 3
  - downloading 47
  - editions 4, 5
  - features 14, 17
  - high-level architecture 13
  - installation errors 197-204
  - installation, preparing for 46
  - logging 204
  - monitoring 204
  - references 208
  - requisites, for installation 46
- vRealize Automation Advanced edition 6**
- vRealize Automation Appliance**
  - about 11
  - configuring 52-54
- vRealize Automation Application Services 6.2 software**
  - downloading 117
- vRealize Automation Application Services Appliance**
  - configuring 119, 120
  - installing 118
- vRealize Automation architecture**
  - advanced services 38
  - approval policies 38
  - blueprints 35
  - Business group 33
  - catalog management 37
  - compute resources 33, 35
  - credentials 34
  - design considerations 33
  - designing 29-32
  - endpoints 34
  - Fabric groups 34
  - naming conventions 33
  - reservations 34, 35
  - users and groups 36
- vRealize Automation Enterprise edition 7**
- vRealize Automation guest agents 90, 91**
- vRealize Automation Identity Appliance**
  - about 11
  - configuring 50-52
  - installing 48, 49
  - upgrading 63, 64
- vRealize Automation Standard edition 6**
- vRealize Automation Virtual Appliance**
  - installing 49, 50
- vRealize Automation web portal**
  - overview 187-190
- vRealize Operations Manager 6.0**
  - about 176
  - benefits, of integrating with vRealization Automation 177
  - integrating, with vRealization Automation 177-179
  - overview 176
  - utilizing, to reclaim idle virtual machines 180-184
- vRealize Operations (vROps) 4**
- vRealize Orchestration (vRO)**
  - about 21
  - prerequisites, configuring 140-148

## **vRealize Orchestrator Client**

- Administer mode 139
- Design mode 138, 139
- navigating through 138, 139
- Run mode 138

## **vRealize Orchestrator configuration**

- about 133, 134
- vRealize Orchestrator Client,  
    running 136, 137
- vRO endpoint, configuring in vRA  
    web portal 135

## **vROps integration 15, 16**

# **W**

## **WIM Blueprint**

- about 91
- creating 91, 92

## **Windows Automation Kit (WAK) 91**

## **Windows Deployment Services (WDS) 18, 91**

## **Windows Image Management (WIM) 6**

## **Windows Management Instrumentation (WMI) 12**

## **workflows**

- adding, to vRA web portal 173

## **workflow tools 174**