



Microsoft System Center

Designing Orchestrator Runbooks

David Ziembicki • Aaron Cushner • Andreas Rynes
Mitch Tulloch, Series Editor

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright 2013 © Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number: 2013948711
ISBN: 978-0-7356-8298-6

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at <http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Anne Hamilton

Developmental Editor: Karen Szall

Project Editor: Karen Szall

Editorial Production: Diane Kohnen, S4Carlisle Publishing Services

Cover Illustration: Twist Creative • Seattle

Cover Design: Microsoft Press Brand Team

Contents

<i>Introduction</i>	<i>xi</i>
Chapter 1 Introducing System Center 2012	1
System Center Virtual Machine Manager	2
System Center Operations Manager	2
System Center Service Manager	2
System Center Data Protection Manager	3
System Center Configuration Manager	3
System Center Orchestrator	4
Chapter 2 System Center Orchestrator	5
Runbook Designer	5
Connections and runbook hierarchy	6
Menu and command bar	6
Runbook design surface	6
Activity list	6
Logging	6
Integration packs	7
Runbook Tester	7
Orchestration console	9
Orchestrator Integration Toolkit	10
Chapter 3 Orchestrator architecture and deployment	13
Architecture	13
System architecture	13
Runbook	13
Management server	13
Runbook server	14
Orchestrator database	14

Runbook Designer	14
Runbook Tester	14
Orchestration console	15
Orchestrator web service	15
Deployment Manager	15
Data bus	16
Architectural diagram	16
High availability considerations	17
Management server	17
Orchestration database	17
Orchestrator web service	18
Orchestration console	18
Runbook servers	18
Runbooks	19
Orchestrator 2012 architecture patterns	19
Single-server Orchestrator 2012 infrastructure	19
High availability Orchestrator 2012 infrastructure	20
Orchestration database	21
Runbook servers	21
Orchestrator web service	21
High availability and multisite Orchestrator 2012 infrastructure	22
Chapter 4 Modular runbook design and development	25
What is a runbook?	25
Creating runbooks	25
Runbook Designer	26
Runbook properties	26
Runbook permissions	27
Using runbook activities	27
Standard activities	27
Monitoring activities	27

Customized activities	28
Common activity properties.....	28
Controlling runbook workflow execution	28
Starting point.....	28
Links.....	29
Loops.....	29
Invokes.....	30
Orchestrator data bus.....	30
Return data activities	31
Extend functionality with integration packs.....	32
Microsoft-provided integration packs.....	32
Third-party integration packs.....	33
Community-developed integration packs	33
Modular runbook design.....	33
Modular management architecture	33
Automation layer	35
Management layer	35
Orchestration layer	35
Runbook design fundamentals.....	36
Error handling	37
Logging	39
Runbook activity pattern.....	39
Modular runbook architecture.....	41
Component runbooks	41
Control runbooks	42
Initiation Runbooks	42
Developing a systematic approach to IT process automation	43
Runbook requirements gathering	44
Process mapping and optimization.....	44
Documenting runbook functional specifications	46

Runbook authoring and development.....	48
Runbook testing.....	48
Runbook versioning and management.....	49
Naming.....	49
Folder structure.....	49
Component runbooks.....	49
Control runbooks.....	50
Initiation runbooks.....	50
Sample of Orchestrator structure.....	50
Runbook versioning.....	51
Component runbooks.....	51
Control runbooks.....	53
Initiation runbooks.....	53
Storing version information.....	53
Microsoft Team Foundation Server integration.....	54
Runbook deployment and monitoring.....	54

Chapter 5 Orchestrator runbook best practices and patterns 55

Runbook design best practices.....	55
Flow control.....	55
Publishing data.....	57
Logging execution data.....	58
Looping.....	59
Sequential vs. parallel activity execution.....	60
Setting job concurrency.....	61
Using Windows PowerShell in Orchestrator.....	61
Windows PowerShell remoting.....	62
Subscribe to Published Data.....	62
Set trace and status variables to defaults.....	62
Validate inputs.....	63
Establish PS remote session.....	63

Execute script in remote session.....	63
Use try/catch/finally.....	63
Append useful data to the Trace variable.....	64
Add any required Windows PowerShell modules.....	64
Use throw for common errors.....	65
Perform core task logic.....	65
Set ErrorState and ErrorMessage.....	65
Return results.....	66
Prep data for Orchestrator Publishing.....	66
Close remote session.....	66
Putting it all together.....	66
Returning arrays.....	68
Runbook patterns.....	70
Component runbook pattern.....	70
Rules.....	71
Error handling.....	72
Validation of input parameters.....	72
Range validation (1-12):.....	73
Enum validation (blue, red):.....	73
Email address validation:.....	73
Date validation:.....	73
IP address validation:.....	73
Control runbook pattern.....	73
Rules.....	74
Error handling.....	75
Validation of input parameters.....	76
Connectivity runbook.....	76
Initiation runbooks.....	79
Rules.....	79
Error handling.....	79

Service requests initiation runbooks.....	80
File-based runbook variables.....	82
Local runbook variables.....	83
Shared runbook variables.....	85
File-based state pattern.....	87
Working with XML in Windows PowerShell.....	88
Creating the file.....	88
Non-exclusive read/write.....	88
Exclusive read/write.....	88
File open pattern for file locks.....	89
Delete file.....	89
Local state pattern.....	90
Shared state and routing/engine patterns.....	90

Chapter 6 Modular runbook example 95

Requirements.....	95
VM provisioning input XML file.....	96
Component runbooks.....	97
Component runbook detail: GetVMHostRating.....	98
Initialize Data.....	99
Get VM Host Rating.....	99
Return Data.....	105
Link and Conditional Logic.....	105
Component runbook summary.....	107
Component runbook detail: CreateNewVM.....	107
Initialize Data.....	108
Create New VM.....	109
Return Data.....	117
Links and Conditional Logic.....	117
Control runbooks.....	117
Control runbook detail: Monitor VM Provisioning.....	117

Control runbook: VM Provisioning Engine.....	118
Remaining control runbooks.....	124
Initiation runbook.....	124
Initiation runbook: Initiate VM Provisioning.....	124
Chapter 7 Calling and executing Orchestrator runbooks	127
Orchestration console.....	127
Orchestrator REST API.....	130
Microsoft Visual Studio.....	130
Windows PowerShell.....	133
System Center Service Manager service catalog.....	136
Create an initiation runbook.....	137
Create an Orchestrator connector.....	138
Create a runbook automation activity template.....	143
Create a service request template.....	146
Create a request offering.....	149
Create a service offering.....	155
Appendix A Windows PowerShell source code for core component runbooks	159
Get Runbook Path.....	159
Get Relative Folder.....	161
Appendix B Steps to set up VMM to Service Manager integration	163
Management packs.....	163
Create an Operations Manager CI Connector.....	163

Introduction

Welcome to *Microsoft System Center: Designing Orchestrator Runbooks*. We believe that orchestration and automation are becoming increasingly important in IT organizations of all sizes and across all infrastructure types ranging from on-premises to cloud-based. Orchestration and automation can help reduce the cost of IT while improving consistency and quality of IT service delivery. Like any powerful technology, however, it can be both used and abused.

Our objective with this book is to provide a framework for runbook design and IT process automation to help you get the most out of System Center Orchestrator 2012 and to help you utilize Orchestrator in concert with the rest of the System Center for an enterprise-wide and systematic approach to process automation. We will provide detailed guidance for creating what we call “modular automation” where small, focused pieces of automation are progressively built into larger and more complex solutions. We detail the concept of an automation library, where over time enterprises build a progressively larger library of interoperable runbooks and components. Finally, we will cover advanced scenarios and design patterns for topics like error handling and logging, state management, and parallelism. But before we dive into the details, we’ll begin by setting the stage with a quick overview of System Center 2012 Orchestrator and deployment scenarios.

About the companion content

The companion content for this book consists of Windows PowerShell scripts and other code samples. It can be downloaded from the following page:

<http://aka.ms/SCrunbook/files>

Errata & book support

We’ve made every effort to ensure the accuracy of this content and its companion content. Any errors that have been reported since this content was published are listed on our Microsoft Press site:

<http://aka.ms/SCrunbook/errata>

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at mspinput@microsoft.com.

Please note that product support for Microsoft software is not offered through the addresses above.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

<http://aka.ms/tellpress>

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter:

<http://twitter.com/MicrosoftPress>

Orchestrator architecture and deployment

This chapter covers the system architecture of System Center 2012 Orchestrator, the various components that are part of the product and how to deploy it in an environment. It discusses the different options that are available to deploy Orchestrator, the scenarios and requirements that you have to think about first before getting started, and details the high availability concepts for each of the components. We also describe the data bus, a key concept of Orchestrator that will help you build your runbooks and allows you to pass data between activities and runbooks without writing code.

Architecture

This section introduces the system components of Orchestrator, their purpose, and how they are going to work together.

System architecture

The system architecture of Orchestrator contains different components—some of them are required, some of them are optional depending on the requirements and usage. This topic will provide an overview of Orchestrator and describe the components and its capabilities.

Runbook

A runbook is the visual representation of your workflow. You will use a graphical interface, the Runbook Designer, to design and build the IT process automation and create those runbooks. A runbook consists of activities and colorful icons, each performing a specific task (for example, create an alert in System Center Operations Manager or run a Windows PowerShell script) that are linked together. The links between activities provide options to do filtering and give you the ability to use multiple paths and conditional logic such as if-then-else.

Management server

The Orchestrator Management server is the layer that is responsible for the communication between the designer of runbooks (Runbook Designer) and the database that holds all required data such as runbook definitions, instances of runbooks, etc.

Runbook server

A runbook server controls the running instances of a runbook; it directly communicates with the database without using the management server. The Runbook server is a Windows service that can be deployed using the Orchestrator Deployment Manager. You can have multiple runbook servers in your environment to distribute load and increase the total capacity of your automation architecture. It is also used to provide high availability.

Orchestrator database

The Orchestrator database utilizes Microsoft SQL Server and contains all runbook definitions, various logging information, as well as configuration data for Orchestrator. It is a single database that holds all the information required for the entire Orchestrator deployment.

Runbook Designer

The Runbook Designer shown in Figure 3-1 is a tool that is part of Orchestrator which you can use to build, edit, and manage your runbooks. It consists of a design pane and a folder structure to organize your runbooks as well as a toolbox that provide the different activities that you will use to build your workflows.

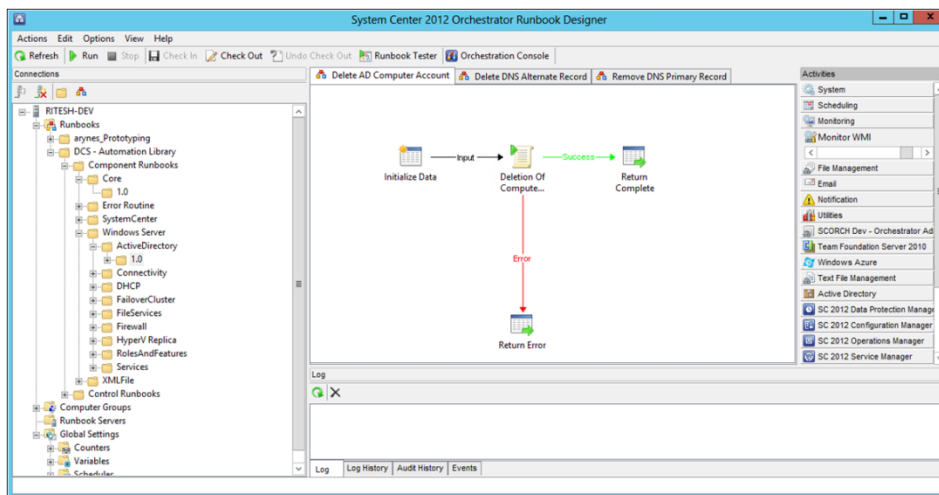


FIGURE 3-1 The System Center 2012 Orchestrator Runbook Designer.

Although the left side of the window shows a similar interface to File Explorer, all runbooks, regardless of how you manage and group them, are stored in the Orchestrator database and not in the file system.

Runbook Tester

The Runbook Tester is another tool that comes with Orchestrator that will help you to test your runbooks at run-time. It will provide you with detailed information and data that are part of

your runbook and is very helpful in troubleshooting problems that you might facing during implementation.

Orchestration console

The Orchestration console lets you manage your runbooks in real-time. You are able to start and stop them and gather real-time status information about current running instances of your runbooks. The tool is a web browser interface that can be accessed remotely.

Orchestrator web service

The Orchestrator web service interface is a Representational State Transfer (REST)-based service that enables custom applications to connect to Orchestrator. It will help to integrate the runbook infrastructure to other portals and tools and give you the option to manage your runbooks such as starting or stopping them or retrieving information about runbook operations.

The Orchestration console uses the Orchestrator web service to communicate with Orchestrator.

Deployment Manager

If you want to use integration packs (IPs) that extend the capabilities of Orchestrator, such as integration of other platforms and tools (for example, HP Service Manager), you need to register and deploy them into your Orchestrator environment. This can be done using the Deployment Manager, a tool that is used to deploy runbook servers and Runbook Designers. The tool is shown in Figure 3-2.

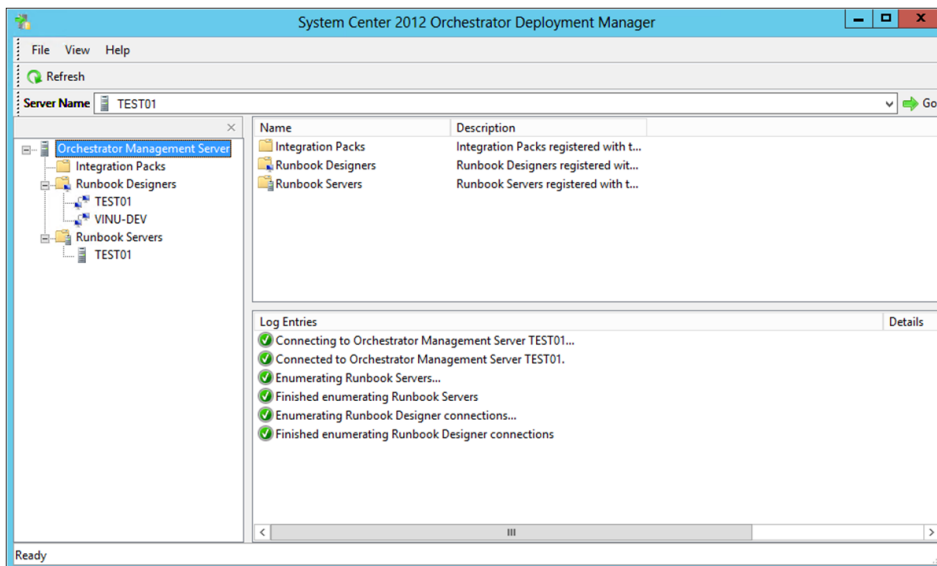


FIGURE 3-2 The System Center 2012 Orchestrator Deployment Manager.

Data bus

Automation is all about moving data between components, platforms, and tools. In most cases you take input data, work with that data, pass it to different systems, get some results back, and at the end of the runbook it's common to return data such as the result state and, optionally, error messages.

The data bus is a key concept of Orchestrator that will help you to do exactly that—pass that data around without writing any code.

To achieve that, the data bus uses another core concept called Published Data. Every step (called activities in Orchestrator) within a runbook will publish data to the data bus automatically. You can also easily add additional data to the data bus. Every subsequent activity in the runbook can access that data bus, subscribe to it, and use the Published Data from there again without any coding.

This makes it very easy to work with data across different vendors and tools to provide automated processes, and it will also decrease the time of implementing those processes significantly.

Architectural diagram

The architectural diagram shown in Figure 3-3 illustrates each of the Orchestrator features and the communication between each component.

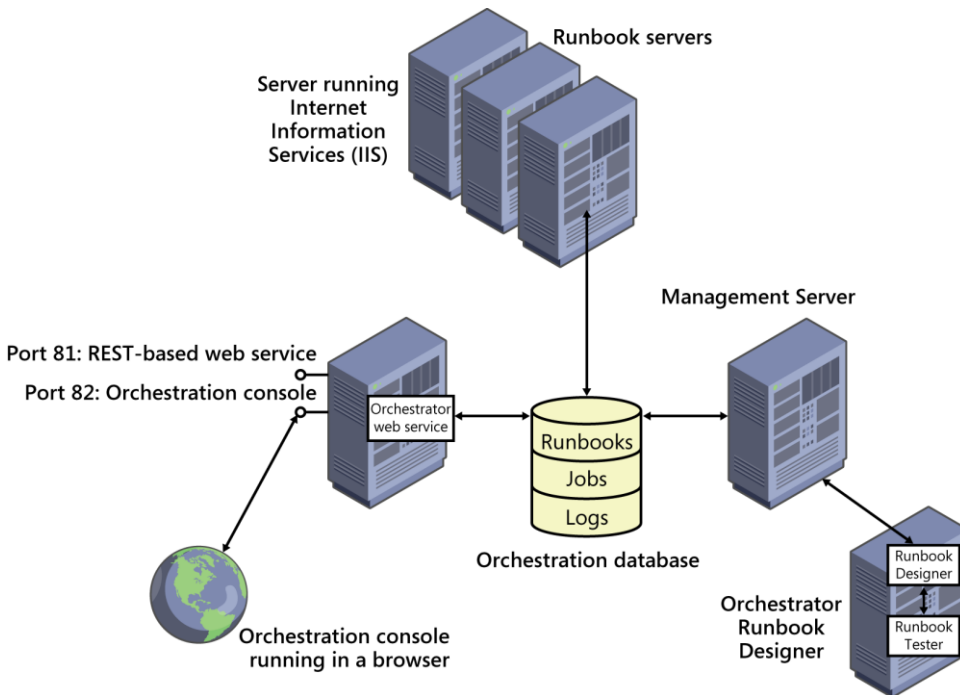


FIGURE 3-3 The Orchestrator system architecture.

The center of every Orchestrator environment is the orchestration database running on Microsoft SQL Server. This is really one of the key critical components of an automation environment using Orchestrator. The database will contain all your runbooks, configuration settings, and logs.

You also need at least one (or more) runbook servers that will communicate directly with the database to retrieve runbooks, to run and store information about the jobs created from the runbooks. So this is the second critical key component that must be available to run your automation processes.

The web services layer also communicates directly with the orchestration database and provides a web browser connection for the Orchestration console. The unavailability of the web service won't impact your automation solution as long as you don't rely on that web service within your implemented runbooks.

The management server is required as a communication layer between the Runbook Designer and the orchestration database, but it is not required for actually running the runbooks, which means a management server that is not available doesn't impact the runbooks from doing their work, but it will impact building new runbooks or changing existing runbooks.

High availability considerations

If you're going to implement automation it will always be the heart of each datacenter and reliability and availability are top most concerns. So this section describes—based on the different components of the product—where high availability is important and how to achieve that.

Management server

This component is actually limited to one single management server, so within every Orchestrator environment there can only be one management server. From an availability perspective there is no requirement that the Orchestrator management server must be up and running or available. Even if the management server is down the runbook servers or runbooks will still run and work.

If the management server is not available, there is an impact on the Runbook Designer. The Runbook Designer cannot be used to publish runbooks or start, monitor, or stop runbooks. Instead, the Orchestration console or the web services must be used to start, monitor, and stop runbooks.

Orchestration database

An Orchestrator 2012 database is hosted on Microsoft SQL Server 2008 or SQL Server 2012 with the basic database engine features, so there is no need for additional features. The database is one of two components that are critical to the entire environment. So it's key to

have an available database for Orchestrator. From a high availability perspective we would recommend you to use Failover Clustering and at least two nodes for SQL Server.

Orchestrator web service

The Orchestrator web service must be installed on a server that is running IIS. If you use the Orchestrator web service in your runbooks then high availability might be required and important to your environment. In general, it is used to start, monitor, or stop runbooks, but it does not have to be available for runbook servers or runbooks to function. For high availability we recommend you install the Orchestrator web service on multiple IIS servers and use a load balancer to both provide additional capacity as well as continuous web service support in case of a failure of one of your web servers.

Orchestration console

The Orchestration console must be installed on a server that is running IIS. The console can be used to start, monitor, or stop runbooks and similar to the web service it is not required to be available for the runbook servers or runbooks to function.

In case you want high availability, we would recommend that you install Orchestration console on more than one IIS server configured for load balancing. This will provide you high availability, but also additional capacity for requests that will use the console.

Runbook servers

Runbook servers are required for runbooks to function. If there isn't a runbook server available, no runbooks will be executed. Therefore high availability is required and important for your Orchestrator environment. But note that runbook servers are not designed to run on a cluster node. To achieve high availability in that case, we recommend deploying at least two runbook servers—depending on your environment and workload it might be required or recommended to have even more than two runbook servers. If the primary runbook server for a runbook is unavailable, the runbook can run on another server.

There are actually two mechanisms in place that are combined, the "spill over" mechanism helps spread the load of runbook instances across the existing runbook servers. So the more runbook servers that are available, the more concurrent instances can run in the same environment. The second mechanism will check for the health of a runbook server using a heartbeat signal. As soon as the runbook server is unhealthy (after 3 missed heartbeats, or 45 seconds) the assignment of a runbook instance will be changed to another runbook server. This makes sure that an unavailable runbook server will not bring down the automation environment and other runbook servers can take over.

You might come across the "Runbook Server Monitor" service in your environment. It monitor the health of the runbook servers and generates Orchestrator platform events should a runbook server show signs of problems. It does not provide any other high availability function in the Orchestrator environment.

Runbooks

A runbook doesn't provide high availability; the runbook server must be high available to make sure your workflows will continue to run in case of a failure. By default, each runbook server is configured to run a maximum of 50 runbooks concurrently. You can change that number based on your experience, resource requirements of your runbooks, and the available resources of your runbook servers.

Consider the resource requirements of the runbooks on a particular server and based on that you can change the default value of 50 to another number by using the runbook server Runbook Throttling tool. In cases where the server has a number of runbooks with high resource requirements, you might want to run fewer concurrent runbooks on the runbook server. In cases of more simple runbooks with small or minimal requirements, you might consider increasing the number of concurrent runbooks on the runbook server.

For additional high availability you can deploy multiple runbook servers, which is described earlier in the "Runbook server section.

Orchestrator 2012 architecture patterns

This section discusses some typical design patterns for Orchestrator 2012 deployments with different requirements and environments. It also gives some examples where you might want to use it, but it doesn't offer general recommendations as there are always requirements that might change the architecture.

The following architecture patterns will be discussed:

- Single-server Orchestrator 2012 infrastructure
- High availability Orchestrator 2012 infrastructure
- High availability and multisite Orchestrator 2012 infrastructure

Single-server Orchestrator 2012 infrastructure

This single-server Orchestrator 2012 infrastructure is a basic deployment where all components are deployed either on a single physical machine or on a single virtual machine. This single machine hosts the management server, Orchestration database, runbook server, Runbook Designer, and Orchestration console.

This design pattern shown in Figure 3-4 is fully functional, but does not provide any high availability or redundancy. In case of a failure, Orchestrator will not be able to process runbooks anymore. So you shouldn't deploy this pattern in a production environment, but it can be used for proof of concepts, demos, or development environments.

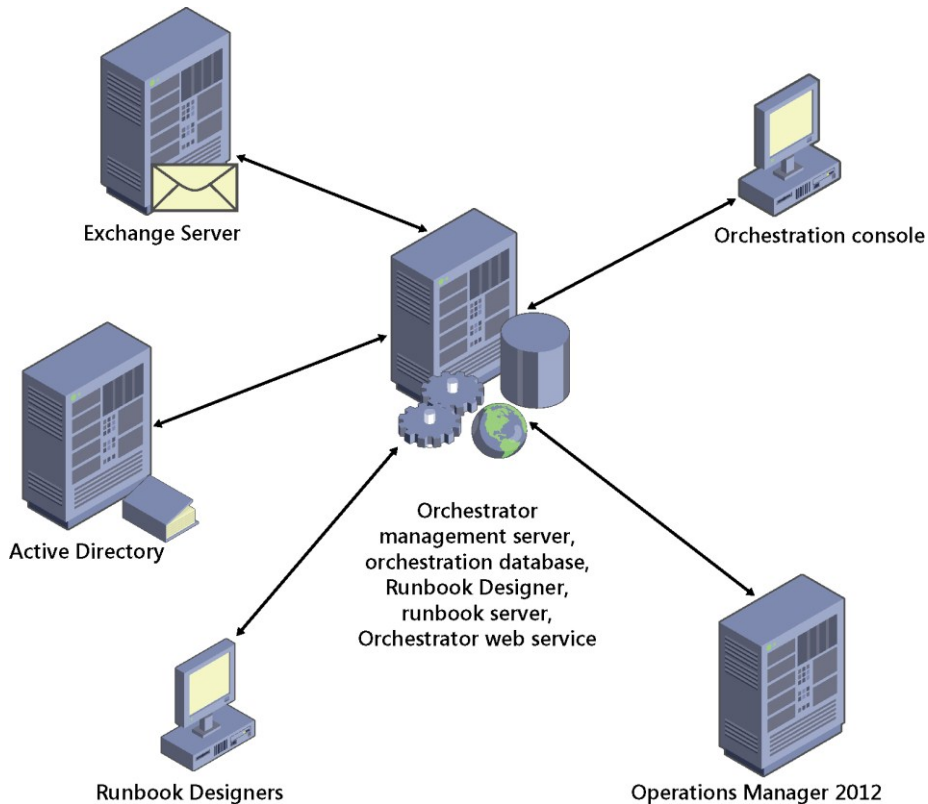


FIGURE 3-4 The Single-server Orchestrator 2012 infrastructure.

Make sure you are aware of the hardware and software requirements. You'll find these at <http://technet.microsoft.com/en-us/library/hh420361.aspx>.

High availability Orchestrator 2012 infrastructure

A high availability Orchestrator 2012 design pattern presented in Figure 3-5 is the most common and widely used one. The critical components of Orchestrator 2012 that were described earlier in this chapter are configured to enable high availability:

- Orchestration database
- Runbook servers
- Orchestrator web service

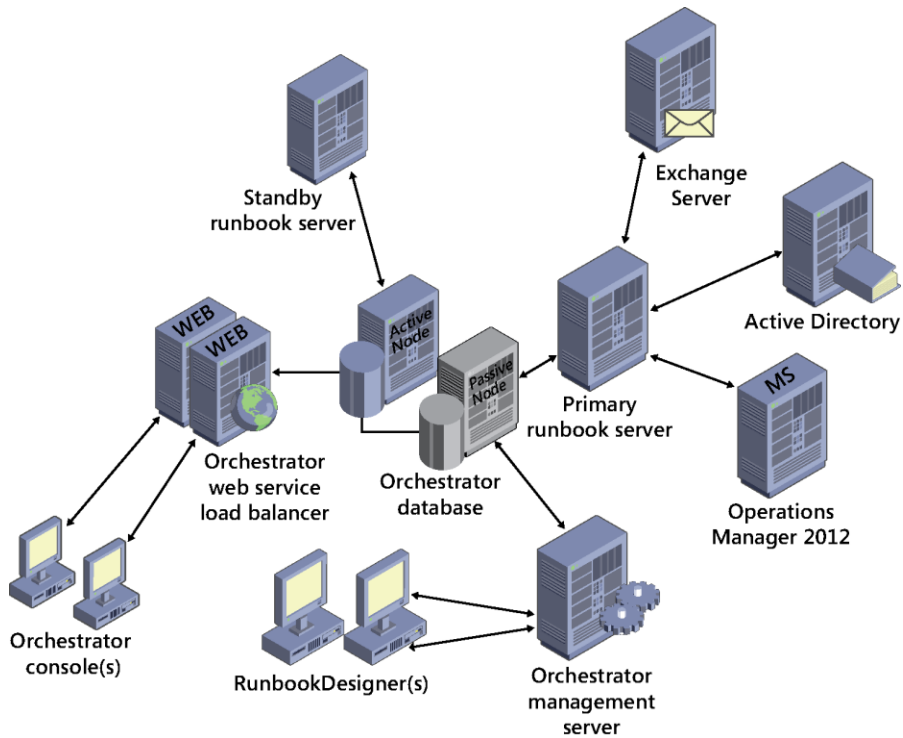


FIGURE 3-5 The High Availability Orchestrator 2012 infrastructure.

Orchestration database

The Orchestration database used to store configuration information, runbooks, and logs is deployed and configured on a two-node SQL Server cluster.

Runbook servers

In Figure 3-5 two runbook servers are deployed for failover and load balancing, but you can use more than two and deploy additional runbook servers to provide additional capacity. The exact number of runbook servers depends on the business and technical requirements of your environment.

Scaling out runbook servers can be performed using the Deployment Tool of Orchestrator, where a wizard guides you through the deployment process in a few steps.

Orchestrator web service

The Orchestration console uses the Orchestrator web service and those two components depend on the performance of the orchestration database and the IIS server that hosts the Orchestrator web service. This is why we would recommend you install the Orchestrator web service on load balanced servers running IIS to provide high availability and additional

capacity. Similar to the runbook servers, the more servers that are available to provide the service, the better the performance and the more requests can be handled.

High availability and multisite Orchestrator 2012 infrastructure

The Orchestrator design pattern presented in Figure 3-6 is intended for managing devices and/or integrating systems from another datacenter in case your network is too slow and/or security plays a more important role than in the other scenarios.

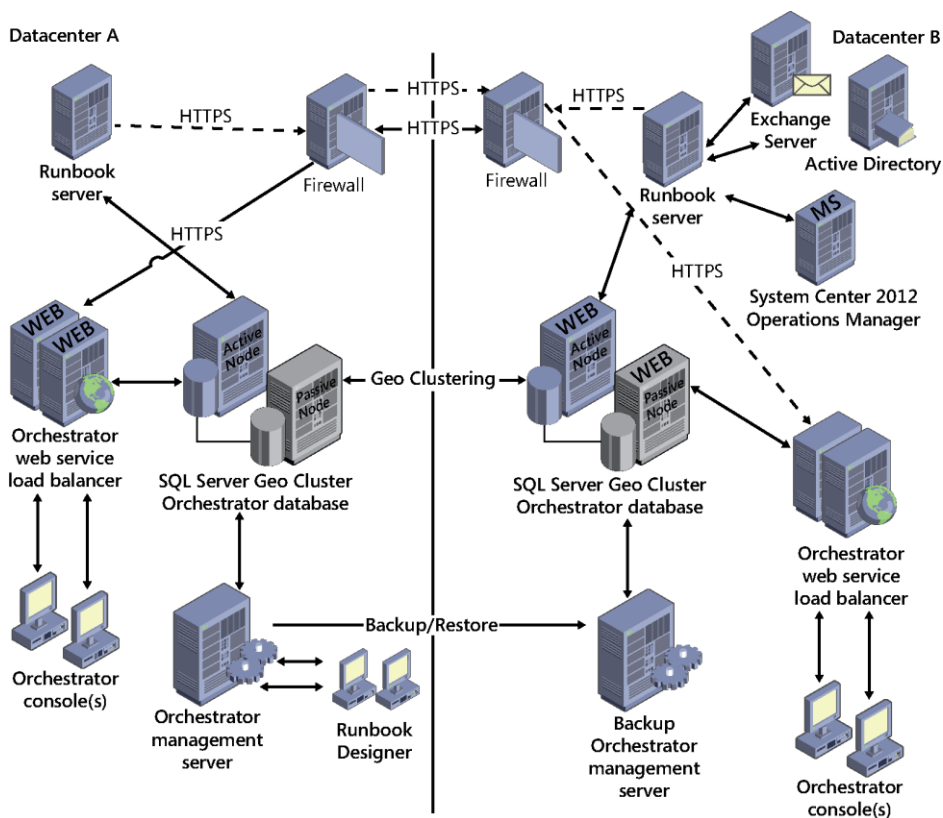


FIGURE 3-6 The High availability and multisite Orchestrator 2012 infrastructure.

On slow networks there is always a risk of packet loss, but configuring a runbook server across a wide area network (WAN) for the purpose of managing devices and/or integrating with systems is not recommended and will result in failures in your runbooks. In cases where latency in your network is greater than 30ms, the recommended approach is to configure a dedicated Orchestrator infrastructure in each datacenter and build runbooks that pass data between the datacenters using the Orchestrator web services. So instead of implementing the

runbooks that will cross datacenter boundaries you will pass the data through the web service interface to another runbook in the remote datacenter.

For SQL you would implement a SQL Server geographically distributed cluster on which you deploy the Orchestrator database. The Orchestrator web services will be deployed separately in each datacenter. Those will access the SQL instance in the same datacenter. Runbook servers will be deployed in each datacenter, each accessing their local SQL instances. If they need to communicate across datacenter sites, the web service of the remote site can be used to leverage https connections.

There is only one active management server, with backups/restores another standby management server is kept updated and in case of a datacenter loss, the backup management server can take over.

