Cisco Intersight

A Handbook for Intelligent Cloud Operations

Matthew Baker | Brandon Beck
Doron Chosnek | Jason McGee | Sean McKeown
Bradley TerEick | Mohit Vaswani

ciscopress.com

Cisco Intersight: A Handbook for Intelligent Cloud Operations

Matthew Baker
Brandon Beck
Doron Chosnek
Jason McGee
Sean McKeown
Bradley TerEick
Mohit Vaswani

Cisco Press

Cisco Intersight: A Handbook for Intelligent Cloud Operations

Matthew Baker
Brandon Beck
Doron Chosnek
Jason McGee
Sean McKeown
Bradley TerEick
Mohit Vaswani

Published by:
Cisco Press

ScoutAutomatedPrintCode

Library of Congress Control Number: 2022914361

ISBN-13: 978-0-13-793728-8
ISBN-10: 0-13-793728-8

Warning and Disclaimer

This book is designed to provide information about operating traditional applications and infrastructure with the agility of cloud-native operations using Cisco Intersight. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Editor-in-Chief: Mark Taub

Director, ITP Product Management: Brett Bartow

Executive Editor: Nancy Davis

Managing Editor: Sandra Schroeder

Development Editor: Ellie C. Bru

Project Editor: Mandie Frank

Copy Editor: Kitty Wilson

Editorial Assistant: Cindy Teeters

Designer: Chuti Prasertsith

Composition: codeMantra

Indexer: Cheryl Lenser

Proofreader: Barbara Mack

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Pearson's Commitment to Diversity, Equity, and Inclusion

Pearson is dedicated to creating bias-free content that reflects the diversity of all learners. We embrace the many dimensions of diversity, including but not limited to race, ethnicity, gender, socioeconomic status, ability, age, sexual orientation, and religious or political beliefs.

Education is a powerful force for equity and change in our world. It has the potential to deliver opportunities that improve lives and enable economic mobility. As we work with authors to create content for every product and service, we acknowledge our responsibility to demonstrate inclusivity and incorporate diverse scholarship so that everyone can achieve their potential through learning. As the world's leading learning company, we have a duty to help drive change and live up to our purpose to help more people create a better life for themselves and to create a better world.

Our ambition is to purposefully contribute to a world where

- Everyone has an equitable and lifelong opportunity to succeed through learning
- Our educational products and services are inclusive and represent the rich diversity of learners
- Our educational content accurately reflects the histories and experiences of the learners we serve
- Our educational content prompts deeper discussions with learners and motivates them to expand their own learning (and worldview)

While we work hard to present unbiased content, we want to hear from you about any concerns or needs with this Pearson product so that we can investigate and address them.

Please contact us with concerns about any potential bias at https://www.pearson.com/report-bias.html.

Figure Credits

Chapter 1, Realistic illustration of a black mobile phone or smartphone with a button, a camera and a blank white screen and a space for a text: Forgem/Shutterstock

Figure 12-7, Logo of Terraform: HashiCorp, Inc.

Chapter 12, Logo of Kubernetes: The Kubernetes Authors

Chapter 12, Logo of Vmware: VMware, Inc.

Chapter 12, Logo of Microsoft Hyper V: Microsoft Corporation

Chapter 12, Logo of Hitachi: Hitachi, Ltd.

Chapter 12, Logo of NetApp: NetApp

Chapter 12, Logo of Pure Storage: Pure Storage, Inc.

Cover, Patrick Hendry/Unsplash

About the Authors

Authors	Titles	Linkedin
Matthew Baker	Technical Solutions Architect	https://www.linkedin.com/in/mattbake
Brandon Beck	Technical Solutions Architect	https://www.linkedin.com/in/brandon-b-70149695
Doron Chosnek	Technical Solutions Architect	https://www.linkedin.com/in/chosnek
Jason McGee	Principal Architect	https://www.linkedin.com/in/jason-mcgee-b6348894/
Sean McKeown	Technical Solutions Architect	https://www.linkedin.com/in/seanmckeown
Bradley TerEick	Technical Solutions Architect	https://www.linkedin.com/in/bradtereick
Mohit Vaswani	Technical Solutions Architect	https://www.linkedin.com/in/mohitvaswani

Dedications

We authors would like to express our deep gratitude for the boundless support and patience our families afforded us during this time-intensive process. We recognize that at times it probably didn't seem worth the burden you had to bear, but we're confident that when you see the royalty checks, you'll know you were right.

We'd also like to thank our dear friends whiskey and beer for their tireless support on countless late nights. Without you, the book would have far fewer typose.

Acknowledgments

We authors would like to express our thanks to Jeremy Foster, Sean McGee, Vance Baran, David Cedrone, and Cynthia Johnson for their unwavering support and encouragement from the initial idea through the final production of this book.

In addition, Bhaskar Jayakrishnan, Sebastien Rosset, Vijay Venugopal, Chandra Krishnamurthy, and Dan Hanson have provided outstanding guidance and expertise for both the business and technical aspects of the topics covered.

Justin Barksdale and Michael Doherty served as technical contributors and reviewers for much of the Kubernetes and infrastructure as code content. Sho Funatsu was a critical reviewer of the storage operations chapter. Prithvi Shankar Ramesh and Subbu Chandrasekaran served as technical reviewers for the virtualization operations chapter and gave us access to their lab. We authors greatly appreciate their time, expertise, and witty interactions.

Throughout the creation of the book, the entire Product Management and Technical Marketing teams for Intersight served as subject matter experts and encouragers. We would like to especially thank David Soper, Jeff Foster, Jeff New, Gregory Wilkinson, Matt Ferguson, Michael Zimmerman, Andrew Horrigan, Vishwanath Jakka, Meenakshi Kaushik, Joost Van Der Made, Jacob Van Ewyk, Gautham Ravi, Matthew Faiello, Chris Atkinson, Ravi Mishra, Chris O'Brien, John McDonough, and Eric Williams.

A special thanks to the world-class Sales, Marketing, and Engineering teams at Turbonomic.com for their longstanding partnership, mentorship, and contributions to the workload optimization content.

Faith Bosworth showed a super-human amount of patience, a wonderful sense of humor, and strong leadership in keeping the authors organized and on task to produce the first edition of this book.

Also, many thanks to Nancy Davis, Ellie Bru, and the numerous Cisco Press staff working behind the scenes to make this second edition happen.

Contents at a Glance

Reader Services

Register your copy at www.ciscopress.com/title/ISBN for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to www.ciscopress.com/register and log in or create an account*. Enter the product ISBN 9780137937288 and click Submit. When the process is complete, you will find any available bonus content under Registered Products.

*Be sure to check the box that you would like to hear from us to receive exclusive discounts on future editions of this product.

Contents

Foreword

Galileo Galilei once said that "passion is the genesis of genius," and I think that perfectly describes the extraordinary group of subject matter experts who came together to author this book; they are passionate geniuses. These seven solutions architects are members of Cisco's Cloud Infrastructure & Software Group and are passionate about the products and technologies available to help our customers successfully build hybrid cloud and multicloud solutions. On a daily basis, they work directly with Cisco customers and partners to combine various ingredients from Cisco, third parties, open sources, and the public cloud to help build exactly what is needed to meet business objectives. The passion these authors have and their decades of combined knowledge and experience make them the perfect team to write this book.

While Cisco has an extensive portfolio of products and solutions across many architectures that can be considered components of a hybrid or multicloud solution, one platform stands out above the rest: Intersight. As these authors describe it, Intersight is a cloud operations platform that delivers intelligent visualization, optimization, and orchestration for applications and infrastructure across multicloud environments. The Intersight platform is *the* core of a successful cloud operations model that must span a customer's infrastructure on premises, colocated, at the edge, or in the public cloud.

This foundational book provides an in-depth overview of the entire Intersight platform, including key concepts, architectural principles, and best practices that will assist organizations as they transition to an intelligent cloud operations model.

M. Sean McGee

Senior Director

Cloud Infrastructure & Software Group

Introduction

Cisco Intersight is a cloud operations platform that delivers intelligent visualization, optimization, and orchestration for applications and infrastructure across multicloud environments. Intersight offers a new paradigm that allows traditional infrastructures to be operated and maintained with the agility of cloud-native infrastructure. In addition, Intersight serves cloud-native environments using many of the proven stability and governance principles inherent in traditional infrastructure.

Getting to this point has been a fascinating journey, especially for a longstanding technology company such as Cisco. The initial development of Intersight and its continued evolution has been a path filled with both exciting innovations and its fair share of challenges to overcome, requiring cultural shifts, partnerships, and an extremely dedicated engineering team.

Many organizations have dealt with the struggles of managing and operating IT infrastructure for years. Over time, there has been little consolidation in the data center space; the number of bare-metal devices has expanded rapidly, and the management challenge has been exacerbated by the exponential increase in storage, network, compute virtualization, and, more recently, containerization. Technologies have been evolving that allow for increased agility and faster response times, but little has been done to decrease complexity. With the rapid adoption of these new technologies, organizational silos have often increased, and there has been no quick solution to ease the management burden. Antithetically, both IT vendors and third-party companies have created more and more tools and consoles to "make life easier." All this has achieved is to add tooling sprawl on top of the already overwhelming technology sprawl. Operations teams have been stretched thin and often have had to divide and conquer to develop the specialized skills to get their work done.

To compensate, IT groups have been broken down into application, security, performance, cloud, network, virtualization, storage, automation, edge, backup, Linux, Windows, and other subteams. Each of these teams typically adopts a suite of tools for their piece of the operations pie. Most tools do not span vendors, let alone account for both virtual and physical servers. Two different paths emerged in data center operations.

The first path was focused on creating a comprehensive view of their environments, and some adventurous operations teams even endeavored to "roll their own" sets of tools and dashboards. These environments consisted of a conglomeration of open-source tools and custom scripts to scrape log files and poll the environment for issues. If correlation happened at all, it was a manual effort, often resulting in many failed attempts to complete root cause analysis. Also, most of these homegrown management systems relied heavily on one or two key individuals in the organization. If those people left the company, the management system usually came crumbling down shortly after their departure.

On the second path, rather than creating their own management tools, organizations began to adopt commercial software packages that were designed to consolidate the vendor-specific or domain-specific tools. Legacy tools such as Microsoft SMS, Tivoli

Management Framework, and HP OpenView were expensive, cumbersome, and rarely fully implemented. These and other similar tools created what was often referred to as the "wedding cake" approach to systems management, with tool upon tool layered in a manager-of-managers approach.

This complexity led many organizations to quickly begin adopting the public cloud after Amazon launched the Elastic Compute Cloud (EC2) service in 2006. Over time, AWS, Google, Azure, and other public cloud providers have built services that are vital for businesses. However, the public cloud has not solved the operational issue; it has relocated the problem and, in many ways, added complexity.

Over the past several years, the industry has seen the rise of AIOps as a buzzword. The concept suggests that IT organizations should use assistive technologies such as machine learning and artificial intelligence to offload burdensome manual tasks, become more efficient, and improve uptime. Tools using these technologies have emerged in the industry, promising improved operations capabilities across private, public, and even hybrid cloud environments.

Cisco identified a major gap between concept and reality concerning true multicloud infrastructure operations, which include not just traditional hardware infrastructures such as servers, storage, and networking but also software resources such as hypervisors, virtual workloads, container services, and public cloud services. Cisco introduced Intersight to address this gap as a true cloud operations platform across myriad infrastructure and cloud services, applying appropriate AIOps techniques to make systems not only easier to manage but more efficient and performant.

As a result, Intersight allows operations teams to:

- Monitor their entire environment, from infrastructure to applications, and gain visibility into their complex interdependencies
- Connect and correlate multiple threads of telemetry from each component to optimize workload resources and assure performance while lowering costs
- Establish consistent environments by orchestrating technical and nontechnical policies across each component

Who This Book Is For

This book is meant primarily for technical IT operators, administrators, managers, and directors. The intent is to provide anyone who is likely to be hands-on with Intersight at some level—whether on a regular basis for technical administrative or programmatic tasks or less frequently for information gathering or dashboard viewing—a means of beginning their journey with Intersight as well as deepening their existing understanding of specific aspects of the platform.

That said, individual chapters (especially Chapters 1 and 2) are likely to be relevant to other business-level persons who are evaluating Intersight prior to purchase or who are leading an Intersight implementation (VP of IT, CIO, CSO, and so on).

How This Book Is Organized

This book is a comprehensive guide to cloud operations provided by Cisco Intersight. The beginning chapters introduce cloud operations as well as the history and purpose of Intersight. Subsequent chapters focus on specific operational topics and related Intersight capabilities. Each chapter flows from high-level business value (the challenge or why) to more advanced detail around the inner workings of Intersight for the given topic. While this book isn't intended to be an exhaustive user manual, it does include high-level walkthroughs of common tasks performed in Intersight as well as navigation tips for finding relevant information for the given topic within the Intersight user interface.

We strongly encourage all readers to start with Chapters 1 and 2 since the concepts in these two chapters are fundamental to all Intersight features and services. After that, you can either continue sequentially or choose individual chapters that cover your specific domain of interest as the subsequent chapters are intended to be consumed atomically.

How This Book Is Structured

Chapter 1, "Intersight Foundations": This chapter describes the architecture of the Intersight platform. The features and capabilities described in subsequent chapters build on this foundation.

Chapter 2, "Security": A cohesive approach to security is paramount in any operations platform. This chapter details Cisco's security philosophy and implementation for Intersight. The principles described here are used to secure the operation of every service described in the remainder of the book.

Chapter 3, "Infrastructure Operations": It is imperative for any operations platform to integrate and communicate with other operations infrastructure. Integration can be as simple as alerting an end user of a detected problem or as robust as consuming published information and communicating to the end user how that published information affects their infrastructure. This chapter describes Intersight's infrastructure operations capabilities, details how Intersight integrates with other operations platforms, and explains how Intersight facilitates typical infrastructure operations functions.

Chapter 4, "Server Operations": Server infrastructure operations is a foundational capability of the Intersight platform. This chapter describes how to deploy, configure, operate, and update Cisco UCS servers with Intersight. This chapter highlights how Intersight enables flexible server configuration practices while ensuring configuration consistency.

Chapter 5, "Network Operations": Cisco UCS offers an integrated network design and an operations experience that is enabled with Intersight. This chapter describes how to deploy, configure, operate, and update the Cisco UCS network infrastructure with Intersight. This chapter highlights how Intersight enables flexible network configuration practices while ensuring configuration consistency.

Chapter 6, "Storage Operations": Storage is a critical component of a complete cloud operations solution. This chapter covers Cisco's hyperconverged offering (HyperFlex) in detail as well as how Intersight provides centralized simplified operations to both hyperconverged and traditional storage.

Chapter 7, "Virtualization Operations": Integration with non-Cisco platforms opens the door for Intersight to act as a control plane for virtualized compute infrastructure, whether on premises or in the cloud. This chapter highlights how Intersight interacts with virtualized infrastructure, and it showcases Intersight's virtualization operations capabilities and supports numerous hypervisor and public cloud environments.

Chapter 8, "Kubernetes": Deploying, maintaining, and monitoring Kubernetes clusters across on-premises and cloud environments is a truly daunting task. This chapter details how Intersight provides enhanced Kubernetes cluster management, simplifies service mesh management, and offers a new custom hypervisor for container-based deployments.

Chapter 9, "Workload Optimization": Ensuring workload performance while also minimizing cost and maximizing utilization is a constant struggle for modern IT organizations. This chapter details how the Workload Optimization service addresses this struggle in real time for both public and private cloud workloads, whether virtualized or containerized.

Chapter 10, "Orchestration": Simplifying redundancy of operations in order to speed time to delivery with a consistent outcome is the desire of many in IT but the accomplishment of few. This chapter describes how orchestration in Intersight lowers the barrier to entry and makes it possible to operationalize repeatable activities across the enterprise.

Chapter 11, "Programmability": This chapter takes a "crawl, walk, run" approach to guiding the reader through programming with Intersight. It contains best practices and many easily digestible examples.

Chapter 12, "Infrastructure as Code": "Code-ifying" infrastructure has a number of benefits, including offering ways to self-document, ensure compliance, and minimize risk. This chapter describes how infrastructure as code can benefit IT organizations and how Intersight approaches this concept with industry-leading standards and integrations.

Chapter 9

Workload Optimization

Introduction

IT operations teams essentially have a prime directive against which their success is constantly measured: to deliver performant applications at the lowest possible cost while maintaining compliance with IT policies.

This goal is thwarted by the almost intractable complexity of modern application architectures—whether virtualized, containerized, monolithic or microservices based, on premises or public cloud, or a combination of them all—as well as the sheer scale of the workloads under management and the constraints imposed by licensing and placement rules. Having a handle on which components of which applications depend on which pieces of the infrastructure is challenging enough; knowing where a given workload is supposed to—or allowed to—run is more difficult still; knowing what *specific decisions* to make at any given time across the multicloud environment to achieve the prime directive is a Herculean task, beyond the scale of humans. As a result, this prime directive is oftentimes met with a brick wall.

Workload Optimizer—a separately licensed feature set within the Intersight platform—aims to solve this challenge through application resource management, ensuring that applications get the resources they need when they need them. Workload Optimizer helps applications perform well while simultaneously minimizing cost (in a public cloud) and optimizing resource utilization (on premises) while also complying with workload policies.

Traditional Shortfalls of IT Resource Management

The traditional method of IT resource management has fallen short in the modern data center. This process-based approach typically involves several steps:

Step 1. Setting static thresholds for various infrastructure metrics, such as CPU or memory utilization

Step 2. Generating an alert when these thresholds are crossed

Step 3. Relying on a human being viewing the alert to:

a. Determine whether the alert is anything to worry about. (What percentage of alerts on any given day are simply discarded in most IT shops? 70%? 80%? 90%?)

b. If the alert is worrisome, determine what action to take to push the metric back below the static threshold.

Step 4. Execute the necessary action and then lather, rinse, repeat.

This approach has significant fundamental flaws.

First, most such metrics are merely proxies for workload performance; they don't measure the health of the workload itself. High CPU utilization on a server may be a *positive* sign that the infrastructure is well utilized and does not necessarily mean that an application is performing poorly. Even if the thresholds aren't static but are centered on an observed baseline, there's no telling whether deviating from the baseline is good or bad or simply a deviation from normal.

Second, most thresholds are set low enough to provide human beings time to react to an alert (after having frequently ignored the first or second notifications), meaning expensive resources are not used efficiently.

Third, and maybe most importantly, this approach relies on human beings to decide what to do with any given alert. An IT administrator must somehow divine from all current alerts not just which ones are actionable but *which specific actions to take*. These actions are invariably intertwined with and will affect other application components and pieces of infrastructure in ways that are difficult to predict. For example:

- A high CPU alert on a given host might be addressed by moving a virtual machine (VM) to another host—but which VM?
- Which other hosts?
- Does that other host have enough memory and network capacity for the intended move?
- Will moving that VM create more problems than it solves?

Multiply this analysis by every potential metric and every application workload in the environment, and the problem becomes exponentially more difficult.

Finally, usually the standard operating procedure is to clear an alert, but, as noted previously, any given alert is not a true indicator of application performance. As every IT administrator has seen time and again, healthy apps can generate red alerts, and "all green" infrastructures can still have poorly performing workloads. A different paradigm is needed, and Workload Optimizer provides such a paradigm.

Paradigm Shift

Workload Optimizer is an analytical decision engine that generates *actions* (recommendations that are optionally automatable in most cases) that drive the IT environment toward a desired state where workload performance is assured and cost is minimized. It uses economic principles (the fundamental laws of supply and demand) in a market-based abstraction to allow infrastructure entities (for example, hosts, VMs, containers, storage arrays) to shop for commodities such as CPU, memory, storage, or network resources.

This market analysis leads to actions. For example, a physical host that is maxed out on memory (high demand) would sell its memory at a high price to discourage new tenants, whereas a storage array with excess capacity would sell its space at a low price to encourage new workloads. While all this buying and selling takes place behind the scenes within the algorithmic model and does not correspond directly to any real-world dollar values, the economic principles are derived from the behaviors of real-world markets. These market cycles occur constantly, in real time, to ensure that actions are currently and always driving the environment toward the desired state. In this paradigm, workload performance and resource optimization are not an either/or proposition; in fact, they must be considered together to make the best decisions possible.

Workload Optimizer can be configured to either recommend or automate infrastructure actions related to placement, resizing, or scaling for either on-premises or cloud resources.

Users and Roles

Workload Optimizer leverages Intersight's core user, privilege, and role-based access control functionality (described in Chapter 1, "Intersight Foundations"). Intersight administrators can assign various predefined privileges specific to Workload Optimizer (see Table 9-1) to a given Intersight role to allow for a division of privileges within an organization. By default, an Intersight administrator has full Administrator privileges in Workload Optimizer, and an Intersight read-only user is granted Observer privileges. Other Workload Optimizer privileges (for example, WO Advisor, WO Automator) must be explicitly assigned to a role via Settings > Roles.

Table 9-1 *Privileges and Permissions*

Workload Optimizer Privileges	Permissions
Workload Optimizer Observer	Can view the state of the environment and recommended actions. Cannot run plans or execute any recommended actions.
Workload Optimizer Advisor	Can view all Workload Optimizer charts and data and run plans. Cannot reserve workloads or execute any recommended actions.
Workload Optimizer Automator	Can execute recommended actions and deploy workloads. Cannot perform administrative tasks.

Workload Optimizer Deployer	Can view all Workload Optimizer charts and data, deploy workloads, and create policies and templates. Cannot run plans or execute any recommended actions.
Workload Optimizer Administrator	Can access all Workload Optimizer features and perform administrative tasks to configure Workload Optimizer.

Targets and Configuration

For Workload Optimizer to generate actions, it needs information to analyze. It accesses the information it needs via API calls to targets, as configured under the Admin tab (refer to Chapter 1). The information gathered from infrastructure targets—metadata, telemetry, and metrics—must be both current and *actionable*.

The number of possible data points available for analysis is effectively infinite, and Workload Optimizer gathers only data that has the potential to lead to or impact a *decision*. This distinction is important as it can help explain why a given target is or is not available or supported. In theory, anything with an API could be integrated as a target, but the key question would be "What decision would Workload Optimizer make *differently* if it had this information?"

One of the great advantages of this approach—and the economic abstraction that underpins the decision engine—is that it scales. Human beings are easily overwhelmed by data, and more data usually just means more noise that confuses the situation. In the case of Workload Optimizer's intelligent decision engine, the more data it has from a myriad of heterogeneous sources, the smarter it gets. More data in this case means more signal and better decisions.

Workload Optimizer accesses its targets in three basic ways (see Figure 9-1):

- Making direct API calls from the Intersight cloud to other cloud services and platforms such as Amazon Web Services, Microsoft Azure, and AppDynamics SaaS (that is, directly cloud to cloud)
- Communicating directly with targets that natively run Device Connector
- Via the Assist function of Intersight Appliance, which enables Workload Optimizer to communicate with on-premises infrastructure natively lacking Device Connector (that is, most third-party hardware and software) that otherwise would be inaccessible behind an organization's firewall.

It is therefore possible to use Workload Optimizer as a purely SaaS customer, as a purely on-premises customer, or as a mix of both.

While all communication to targets occurs via API calls, without any traditional agents required on the target side, Kubernetes clusters do require a unique setup step: deploying Kubernetes Collector on a node within the target cluster. Collector runs with a service account that has a cluster administrator role and runs Device Connector, essentially proxying communications to and commands from Intersight and the native cluster kubelet or node agent. In this respect, Collector allows the insertion of Device Connector into any Kubernetes cluster, whether on premises or in the public cloud.

Figure 9-1 *Communication with public cloud services and on-premises resources*

One of the richest sources of workload telemetry for Workload Optimizer comes from application performance management tools such as Cisco's AppDynamics. As noted earlier, the core focus of Workload Optimizer is application *resource* management. However, for an application to truly perform well, it needs more than just the right physical resources at the right time; it also needs to be written and architected well.

AppDynamics provides developers and applications IT teams with a detailed logical dependency map of the application and its underlying services, fine-grained insight into individual lines of problematic code and poorly performing database queries and their impact on actual business transactions, and guidance in troubleshooting poor end-user experiences. Figure 9-2 illustrates the combination of application *performance* management (that is, assuring good code and architecture) and Workload Optimizer's application *resource* management (that is, the right resources at the right time for the lowest cost).

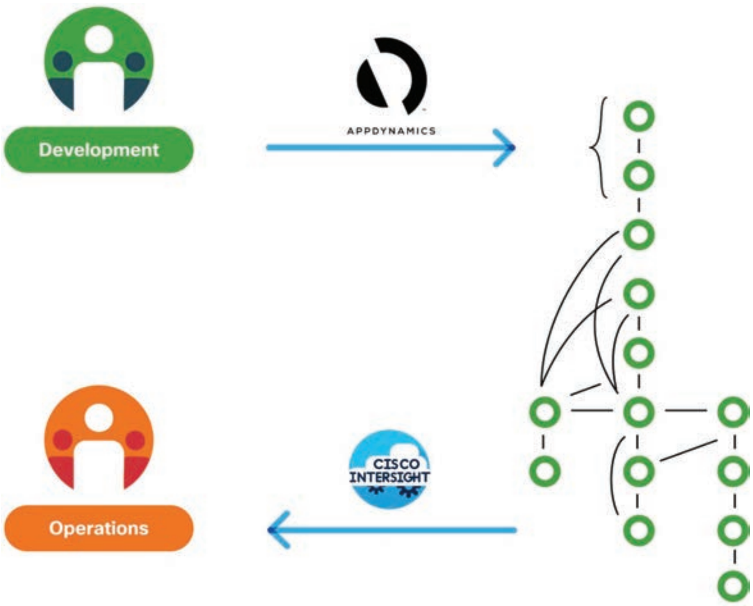

Figure 9-2 *AppDynamics integration into Workload Optimizer*

Cisco Full Stack Observability (FSO) expands on the visibility, insights, and action capabilities of the Workload Optimizer and AppDynamics combination and enhances it with wide area network and end-user monitoring intelligence from Cisco ThousandEyes. The FSO solution currently addresses numerous critical business use cases, such as customer digital experience monitoring and cloud-native application monitoring, and more product integrations and use cases are on the way.

The Supply Chain

Workload Optimizer uses the information it gathers from targets to stitch together a logical dependency mapping of all entities in the customer environment, from the business application at the top to the containerized and/or virtualized workloads in the middle to physical hosts, storage, network, and facilities (or equivalent public cloud services) below. This mapping, called the supply chain (see Figure 9-3), is the primary means of navigation in Workload Optimizer.

The supply chain shows each known entity as a colored ring. The color of a ring indicates the current state of the entity in terms of pending actions—red if there are critical pending performance or compliance actions, orange for prevention actions, yellow for efficiency-related actions, and green if no actions are pending. The center of each ring displays the known quantity of the given entity, and the connecting arrows illustrate consumers' dependencies on other providers.

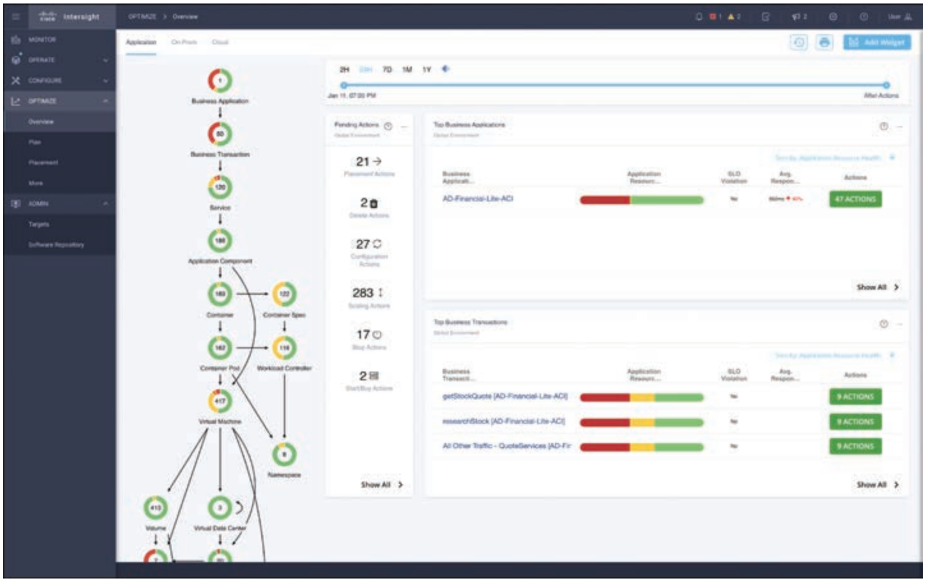

Figure 9-3 *The Workload Optimizer supply chain*

Clicking on a given entity in the supply chain opens a context-specific window with detailed information about that entity type. For example, in Figure 9-4, the Container entity in the supply chain has been selected, and you can see many container-relevant widgets, as well as a series of tabs for policies, actions, and so on.

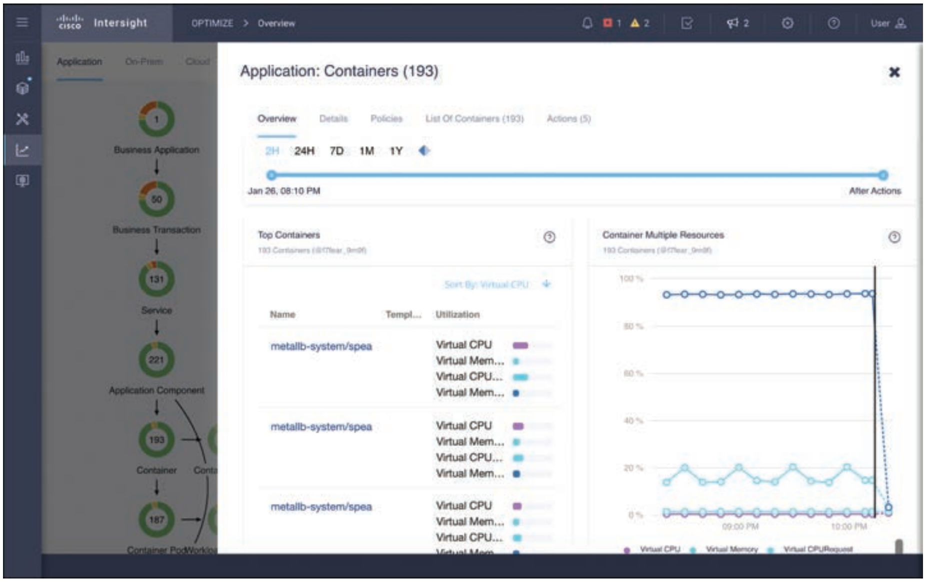

Figure 9-4 *Accessing additional details by clicking in the supply chain*

Furthermore, the supply chain is *dynamic*, meaning that if a particular entity, such as a specific business application (for example, AD-Financial-Lite-ACI in Figure 9-5), is selected, the supply chain automatically reconfigures itself to depict just that single business application and only its related dependencies (including updating the color of the various rings and their respective entity counts). This is known as *narrowing the scope* of the supply chain view and is extremely helpful for focusing on a specific area of concern or work. Clicking on the Home link at the top-left of the Workload Optimizer screen or selecting Optimize > Overview on the main Intersight menu bar on the left returns you to the full scope of the supply chain.

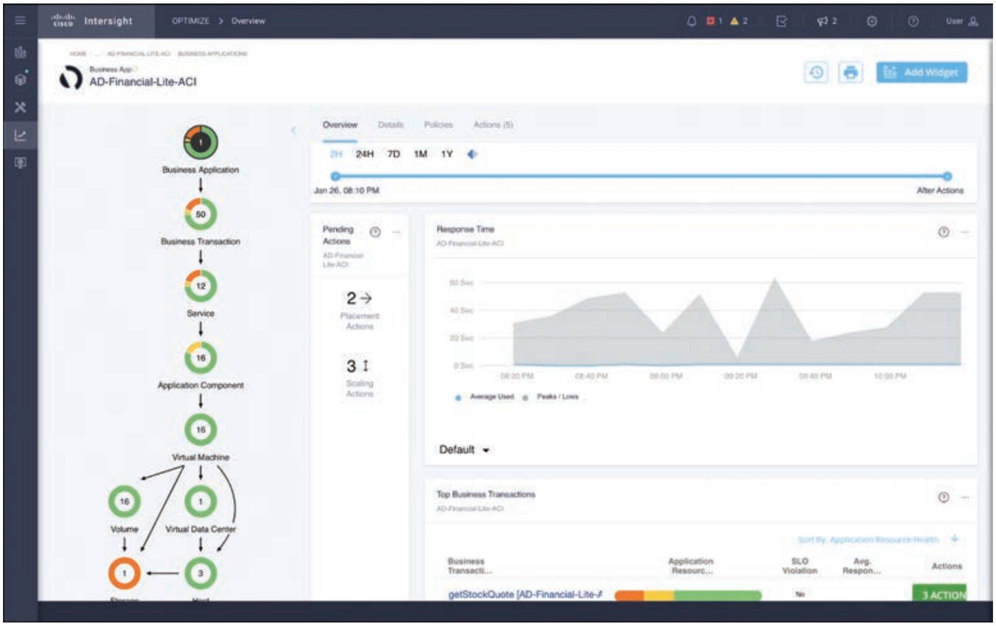

Figure 9-5 *Scoped supply chain view of a single application*

Actions

The essence of Workload Optimizer is *action*. Many infrastructure tools promise visibility, and some even provide some insights on top of that visibility. Workload Optimizer is designed to go further and act, in real time, to continuously drive the environment toward the desired state. Actions run the gamut from placement and scaling actions for workloads and storage to infrastructure start/stop/provision/decommission actions to public cloud purchase recommendations and many more.

Workload scaling and placement actions are often the most common and have the most impact. For VMs, Workload Optimizer leverages the underlying hypervisor to scale up or down resource capacity (for example, memory and CPU), reservations, and limits, as well as move VMs and datastores to nodes or clusters that can more optimally support their needs. Similarly, in Kubernetes environments, Workload Optimizer offers actions to right-size container vCPU and vMem requests and limits, move pods to different nodes to free up resources or avoid congestion, and so on. Keep in mind that the list of supported actions and their ability to be executed or automated via Workload Optimizer varies widely by target type and updates frequently. A current detailed list of actions and their execution support status via Workload Optimizer can be found in the Workload Optimizer Target Configuration Guide (http://cs.co/9006zF6Bi).

All actions follow an opt-in model; Workload Optimizer never takes an action unless given explicit permissions to do so, either via direct user input or through a custom policy. You can view a list of current actions via the Pending Actions dashboard widget in the supply chain view, via the Actions tab after clicking on a component in the supply chain, or in various scoped views and custom dashboards. Figure 9-6 shows an example of a specific move action.

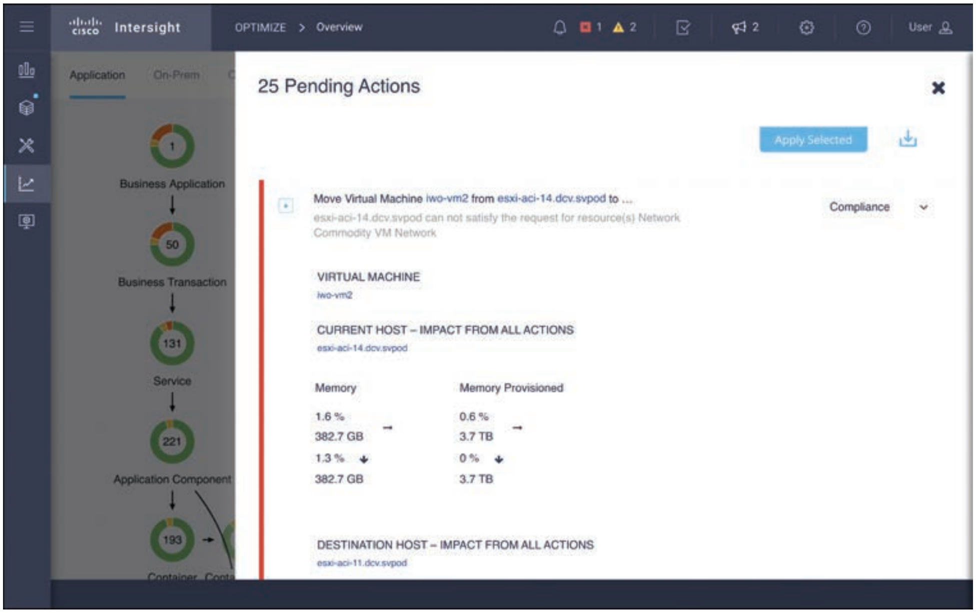

Figure 9-6 *Executing actions*

Groups and Policies

When an organization first starts using Workload Optimizer, the number of pending actions can be significant, especially in a large, active, or poorly optimized environment. New organizations generally take a conservative approach initially and execute actions manually, verifying as they go that the actions are improving the environment and moving them closer to the desired state. Ultimately, though, the power of Workload Optimizer is best achieved through a judicious implementation of groups and policies to simplify the operating environment and to automate actions where possible.

Groups

Workload Optimizer provides the capability of creating logical groups of resources (VMs, hosts, datastores, disk arrays, and so on) for ease of management, visibility, and automation. Groups can be either static (such as a fixed list of a named set of resources) or dynamic. Dynamic groups self-update their membership based on specific filter criteria—a query, effectively—to significantly simplify management. For example, you could create a dynamic group of VMs that belong to a specific application's test environment, and you could further restrict membership of that group to just those running Microsoft Windows (see Figure 9-7, where .* is used as a catchall wildcard in the filter criteria).

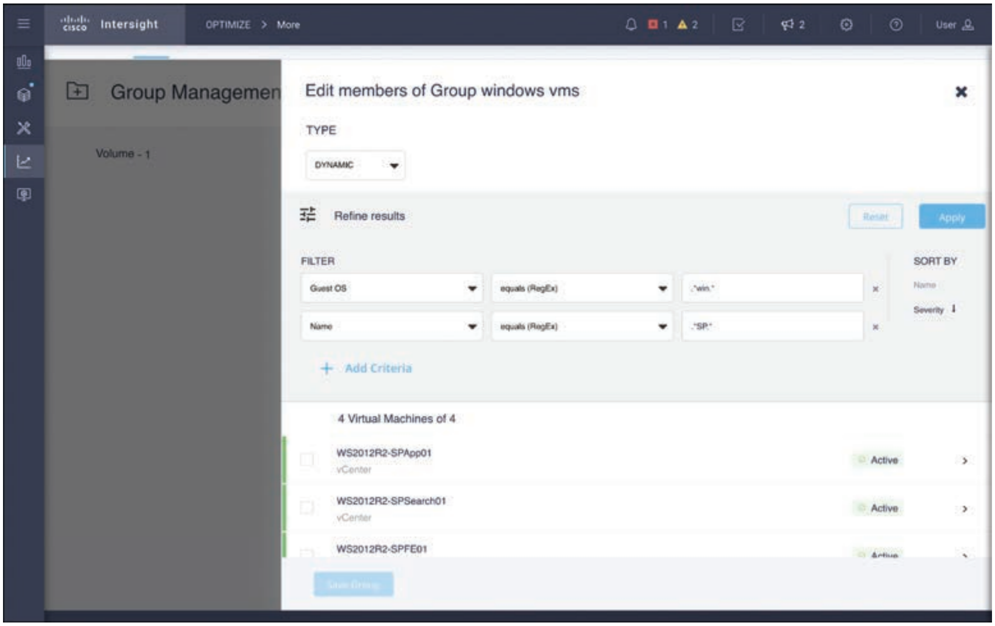

Figure 9-7 *Creating dynamic groups based on filter criteria*

Generally, dynamic groups are preferred due to their self-updating nature. As new resources are provisioned or decommissioned, or as their status changes, their dynamic group membership adjusts accordingly, without any user input. This benefit is difficult to understate, especially in larger environments. You should use dynamic groups whenever possible and static groups only when necessary.

Groups can be used in numerous ways. From the search screen, you can select a given group and automatically scope it to just that group in the supply chain. This is a handy way to zoom in on a specific subset of the infrastructure in a visibility or troubleshooting scenario or to customize a given widget in a dashboard. Groups can also be used to easily narrow the scope of a given plan or placement scenario (as described in the next section).

Policies

One of the most critical benefits of the use of groups arises when they are combined with policies. In Workload Optimizer, all actions are governed by one or more policies, including default global policies, user-defined custom policies, and imported placement policies. Policies provide extremely fine-grained control over the actions and automation behavior of Workload Optimizer.

Policies fall under two main categories: placement and automation. In both cases, groups (static or dynamic) are used to limit the scope of the policy.

Placement Policies

Placement policies govern which consumers (VMs, containers, storage volumes, datastores) can reside on which providers (VMs, physical hosts, volumes, disk arrays).

Affinity/Anti-affinity Policies

The most common use for placement policies is to create affinity or anti-affinity rules to meet business needs. For example, say that you have two dynamic groups, both owned by the testing and development team: one of VMs and another of physical hosts. To ensure that the testing and development VMs always run on testing and development hosts, you can create a placement policy that enables this constraint in Workload Optimizer, as shown in Figure 9-8.

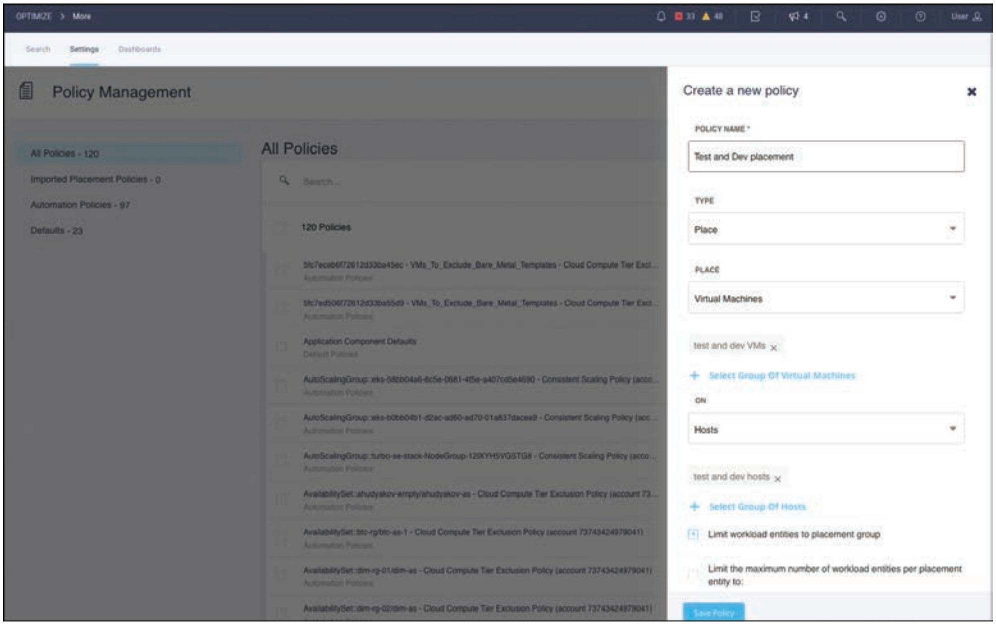

Figure 9-8 *Creating a new placement policy*

The constraint you put into the policy then restricts the underlying economic decision engine that generates actions. The buying decisions that the VMs within the testing and development group make when shopping for resources are restricted to just the testing and development hosts, even if there might be other hosts that could otherwise serve those VMs. You might similarly constrain certain workloads with a specific license requirement to only run on (or never run on) a given host group that is (or isn't) licensed for that purpose.

Merge Policies

Another placement policy type that can be especially useful is a merge policy. Such a policy logically combines two or more groups of resources such that the economic engine treats them as a single, fungible asset when making decisions.

The most common example of a merge policy is one that combines one or more VM clusters such that VMs can be moved between clusters. Traditionally, VM clusters are siloed islands of compute resources that can't be shared. Sometimes this is done for specific business reasons, such as separating accounting for different data center tenants; in other words, sometimes the silos are built intentionally. But many times, they are unintentional: The fragmentation and subsequent underutilization of resources is merely a byproduct of the artificial boundary imposed by the infrastructure and hypervisor. In such a scenario, you can create a merge policy that logically joins multiple clusters' compute and storage resources, enabling Workload Optimizer to consider the best location for any given workload without being constrained by cluster boundaries. This ultimately leads to optimal utilization of all resources in a continued push toward the desired state.

Automation Policies

The second category of policy in Workload Optimizer is automation policies, which govern how and when Workload Optimizer generates and executes actions. Like a placement policy, an automation policy is restricted to a specific scope of resources based on groups; however, unlike placement policies, automation policies can be restricted to run at specific times with schedules. Global default policies govern any resources that aren't otherwise governed by another policy. It is therefore important to use extra caution when modifying a global default policy as any changes can be far-reaching.

Automation policies provide great control—either broad or extremely finely grained control—over the behavior of the decision engine and how actions are executed. For example, it's common for organizations to enable nondisruptive VM resize-up actions for CPU and memory (for hypervisors that support such actions), but some organizations wish to further restrict these actions to specific groups of VMs (for example, testing and development only and not production), or to occur during certain pre-approved change windows, or to control the growth increment. Most critically, automation policies enable supported actions to be executed automatically by Intersight, eliminating the need for human intervention.

Best Practices

When implementing placement and automation policies, a crawl–walk–run approach is advisable.

Crawl

Crawling involves creating the necessary groups for a given policy, creating a policy scoped to those groups, and setting the policy's action to *manual* so that actions are generated but not automatically executed.

This method provides administrators with the ability to double-check the group membership and manually validate that the actions are being generated as expected for only the desired groups of resources. Any needed adjustments can be made before manually executing the actions and validating that they do indeed move the environment closer to the desired state.

Walk

Walking involves changing an automation policy's execution mechanism to automatic for relatively low-risk actions. The most common of these actions are VM and datastore placements, nondisruptive upsizing of datastores and volumes, and nondisruptive VM resize-up actions for CPU and memory.

Modern hypervisors and storage arrays can handle these actions with little to no impact on the running workloads, and automating them generally provides the greatest bang for the buck for most environments. More conservative organizations may want to begin automating a lower-priority subset of their resources (such as testing and development systems), as defined by groups. Combining these "walk" actions with a merge policy to join multiple clusters provides even more opportunity for optimization in a reasonably safe manner.

Run

Finally, running typically involves more complex policy interactions, such as schedule implementations, before- and after-action orchestration steps, and rollout of automation across the organization, including production environments.

During the run stage, it is critical to have well-defined groups that restrict unwanted actions. Many off-the-shelf applications such as SAP have extremely specific resource requirements that must be met to receive full vendor support. In such cases, organizations typically create a group, specific to an application, and add a policy for that group that disables all action generation for it, effectively telling Workload Optimizer to ignore the application. This can also be done for custom applications for which the development teams have similarly stringent resource requirements.

Planning and Placement

While the bulk of functionality built into Workload Optimizer is focused on acting in real time to continuously optimize, there are two additional modules: one that supports future-looking planning and another that supports workload placement.

Plan

Since the entire foundation of Workload Optimizer's decision engine is its market abstraction governed by the economic laws of supply and demand, it is a straightforward exercise to ask *what-if* questions of the engine in the planning module.

The planning function enables users to virtually change either the supply side (by adding or removing providers such as hosts or storage arrays) or the demand side (by adding or removing consumers such as VMs or containers) or both and then simulate the effect of the proposed change(s) on the live environment. Under the hood, this is a simple task for Workload Optimizer because it merely needs to run an extra market cycle with the new (simulated) input parameters. Just as in a live environment, the results of a plan (as shown in Figure 9-9) are *actions*.

Plans answer questions such as:

- If four hosts were decommissioned, what actions would need to be taken to handle the workloads that they are currently running?
- Does capacity exist elsewhere to handle the load, and if so, where should workloads be moved?
- If there is not enough spare capacity, how much more and of what type will need to be bought/provisioned?

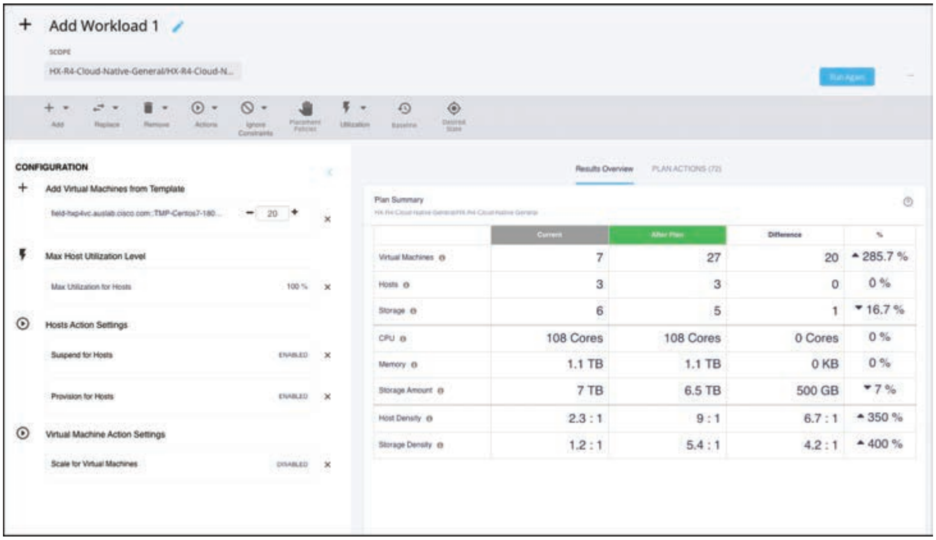

Figure 9-9 *Results of an added workload plan*

The planning in Workload Optimizer takes the concept of traditional capacity planning, which can be a relatively crude exercise in projecting historical trend lines into the future, to a new level: Workload Optimizer does its planning and tells you exactly what actions will need to be taken in response to a given set of changes to maintain the desired state. One of the most frequently used planning types is the Migrate to Cloud simulation, which is addressed in greater detail later in this chapter, in the section "The Public Cloud."

Placement

The placement module in Workload Optimizer is a variation on the planning theme but with real-world consequences. Placement reservations (see Figure 9-10) allow an administrator who knows that new workloads are coming into the environment soon to alter the demand side of all future market cycles, taking the yet-to-be-deployed workloads into account.

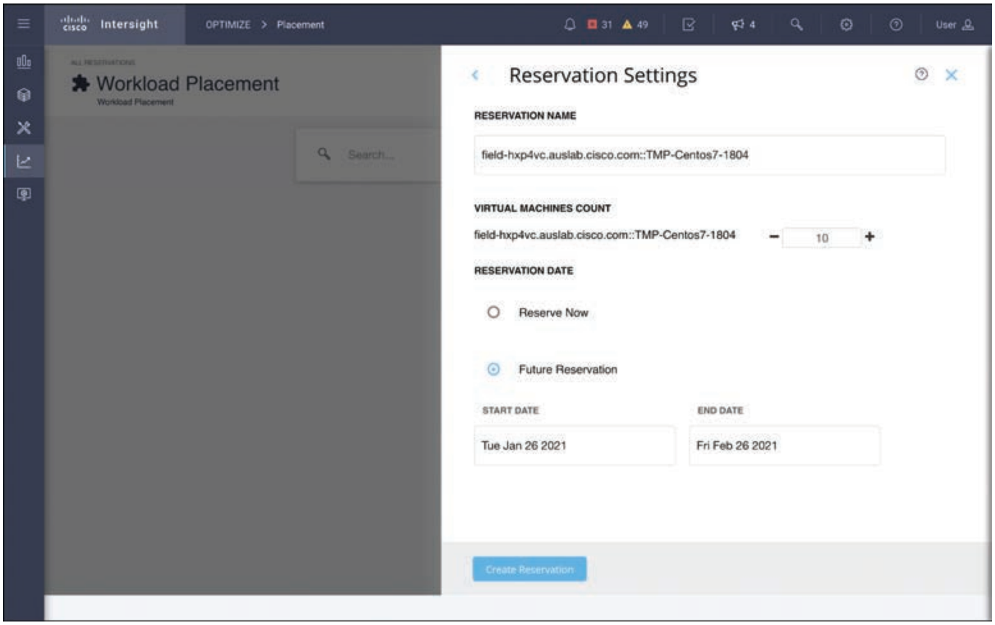

Figure 9-10 *Creating a placement reservation*

Such reservations force the economic engine to behave as if those workloads already exist, and the engine generates real actions accordingly. Reservations may therefore result in real changes to the environment if automation policies are active and/or real recommended pending actions such as VM movements and server or storage provisioning (to accommodate the proposed new workloads) are undertaken.

Using placement reservations is a great way to both plan for new workloads and to ensure that resources are available when those workloads are deployed. A handy feature of any placement reservation is the ability to delay making the reservation active until a point in the future, including the option of an end date for the reservation. This delays the effect of the reservation until a time closer to the actual deployment of the new workloads.

The Public Cloud

In an on-premises data center, infrastructure is generally finite in scale and fixed in cost. By the time a new physical host hits the floor, the capital has been spent and has taken its hit on the business's bottom line. Thus, the desired state in an on-premises environment is to assure workload performance and maximize utilization of the sunk cost of capital infrastructure. In the public cloud, however, infrastructure is effectively infinite. Resources are paid for as they are consumed—usually from an operating expenses budget rather than a capital budget.

The underlying market abstraction in Workload Optimizer is extremely flexible, and it can easily adjust to optimize for the emphasis on operating expenses. In the public cloud, the desired state is to ensure workload performance and minimize spending. This is a subtle but key distinction, as minimizing spending in the public cloud does not always mean placing a workload in the cloud VM instance that perfectly matches its requirements for CPU, memory, storage, and so on; instead, it means placing that workload in the cloud VM template that results in the lowest possible cost while still ensuring performance.

On-Demand Versus Reserved Instances

The public cloud's vast array of instance sizes and types offers endless choices for cloud administrators, all with slightly different resource profiles and costs. There are hundreds of different instance options in AWS and Azure, and new options and pricing are emerging almost daily. To further complicate matters, administrators have the option of consuming instances in an on-demand fashion—that is, in a pay-as-you-use model—or via reserved instances (RIs) that are paid for in advance for a specified term (usually a year or more). RIs can be incredibly attractive as they are typically heavily discounted compared to their on-demand counterparts, but they are not without pitfalls.

The fundamental challenge of consuming RIs is that public cloud customers pay for the RIs *whether they use them or not*. In this respect, RIs are more like the sunk cost of a physical server on premises than like the ongoing cost of an on-demand cloud instance. You can think of on-demand instances as being well suited for temporary or highly variable workloads—analogous to city dwellers taking taxis, which is usually cost-effective for short trips. RIs are akin to leasing a car, which is often the right economic choice for longer-term, more predictable usage patterns (such as commuting an hour to work each day). As the artifact changes, the flexibility of the underlying economic abstraction of Workload Optimizer is up to the challenge.

When faced with myriad instance options, cloud administrators are generally forced down one of two paths: Purchase RIs only for workloads that are deemed static and consume on-demand instances for everything else (hoping, of course, that static workloads really do remain that way) or choose a handful of RI instance types (for example, small, medium, and large) and shoehorn all workloads into the closest fit. Both methods leave a lot to be desired. In the first case, it's not at all uncommon for static workloads to have their demand change over a year (or more) as new end users are added or new functionality comes online. In such cases, the workload needs to be relocated to a new instance type, and the administrator has an empty hole to fill in the form of the old, already paid-for RI (see the examples in Figure 9-11).

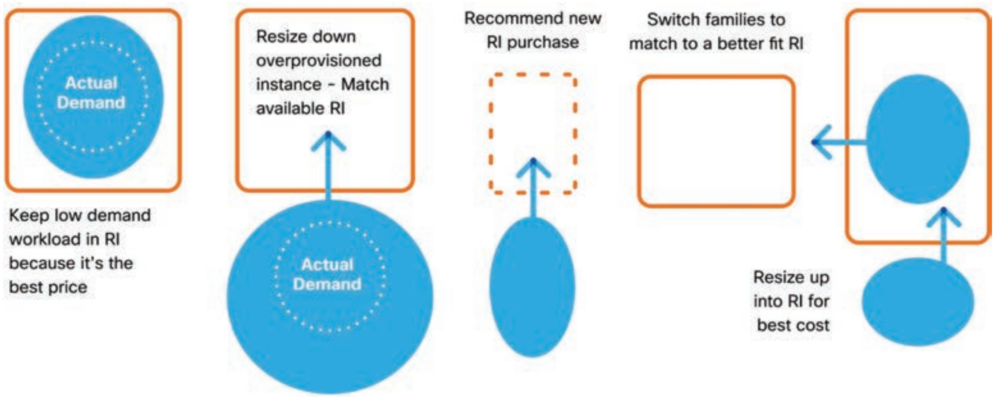

Figure 9-11 *Fluctuating demand creates complexity with RI consumption*

What should be done with that hole? What's the best workload to move into it? Keep in mind that if *that* workload is coming from its own RI, the problem simply cascades downstream. The unpredictability and inefficiency of such headaches often negates the potential cost savings of RIs.

In the second scenario, limiting the RI choices almost by definition means mismatching workloads to instance types, negatively affecting either workload performance or cost savings or both. In either case, human beings, even with complicated spreadsheets and scripts, will invariably get the answer wrong because the scale of the problem is too large, and everything keeps changing all the time—so the analysis done last week or even yesterday is likely to be invalid today.

Thankfully, Workload Optimizer understands both on-demand instances and RIs in detail through its direct API target integrations. Workload Optimizer constantly receives real-time data on consumption, pricing, and instance options from cloud providers, and it combines this data with knowledge of applicable customer-specific pricing and enterprise agreements to determine the best actions available at any given point in time (see Figure 9-12).

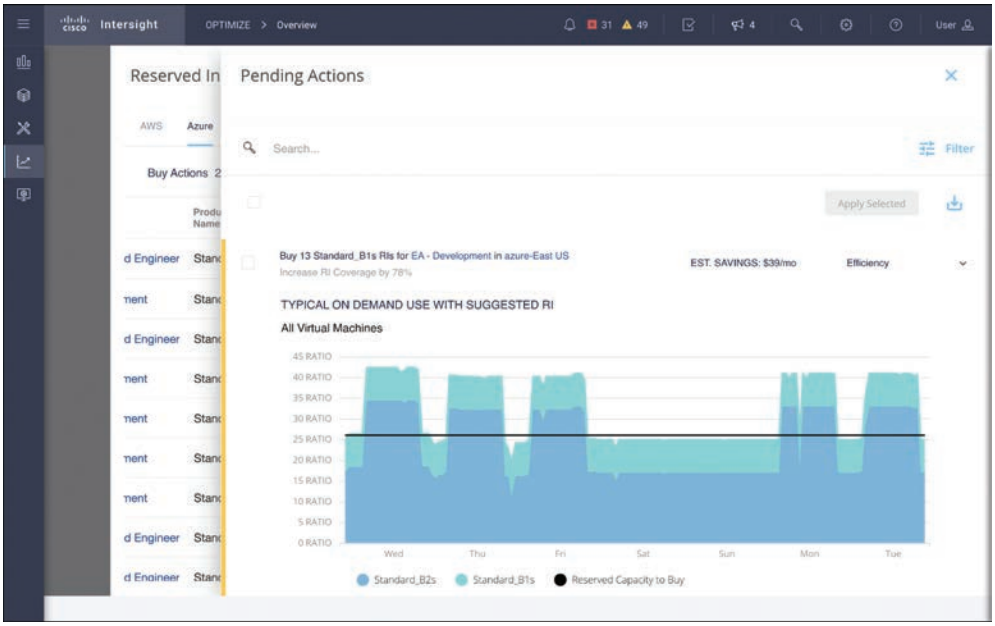

Figure 9-12 *A pending action to purchase additional RI capacity in Azure*

Not only does Workload Optimizer understand current and historical workload requirements and an organization's current RI inventory, but it can also intelligently recommend the optimal consumption of existing RI inventory and recommend additional RI purchases to minimize future spending. Continuing with the previous car analogy, in addition to knowing whether it's better to pay for a taxi or lease a car in any given circumstance, Workload Optimizer can even suggest a car lease (RI purchase) that can be used as a vehicle for ride sharing (that is, fluidly moving on-demand workloads in and out of a given RI to achieve the lowest possible cost while still ensuring performance).

Public Cloud Migrations

Finally, because Workload Optimizer understands both the on-premises and public cloud environments, it can bridge the gap between them. As noted in the previous section, the process of moving VM workloads to the public cloud can be simulated with a plan and the selection of specific VMs or VM groups to generate the optimal purchase actions required to run the workloads (see Figure 9-13).

The plan results offer two options: Lift & Shift and Optimized. The Lift & Shift column shows the recommended instances to buy and their costs, assuming no changes to the size of the existing VMs. The Optimized column allows for VM right-sizing in the process of moving to the cloud, which often results in a lower overall cost if current VMs are oversized relative to their workload needs. Software licensing (for example, bring your own versus buy from the cloud) and RI profile customizations are also available to further fine-tune the plan results.

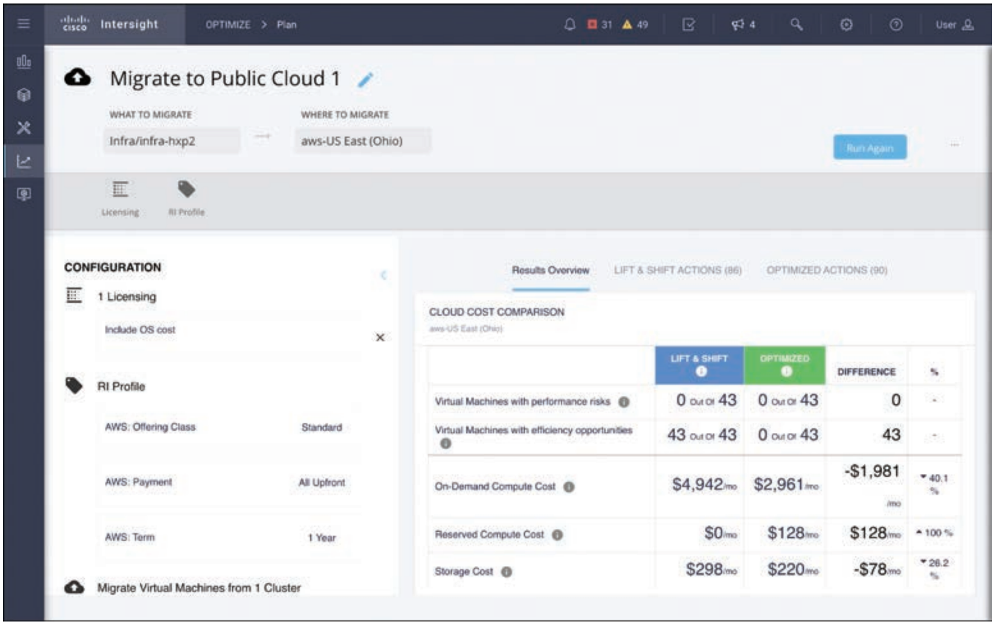

Figure 9-13 *Results of a cloud migration plan*

Workload Optimizer's unique ability to apply the same market abstraction and analysis to both on-premises and public cloud workloads in real time enables it to add value far beyond any cloud-specific or hypervisor-specific point-in-time tools that may be available. Besides being multivendor, multicloud, and real time by design, Workload Optimizer does not force administrators to choose between performance assurance and cost/resource optimization. In the modern application resource management paradigm of Workload Optimizer, performance assurance and cost/resource optimization are blended aspects of the desired state.

Summary

The flexibility and extensibility of the Intersight platform enable the rapid development of new features and capabilities. Additional hypervisor, application performance management, storage, and orchestrator targets are under development, as are additional reporting and application-specific support. Organizations find that the return on investment for Workload Optimizer is rapid as the cost savings it uncovers in the process of assuring performance and optimizing resources quickly exceed the license costs.

References

- Cisco Intersight Help Center, "Cisco Intersight Workload Optimizer Getting Started Guide," https://intersight.com/help/saas/resources/cisco_intersight_workload_optimizer_getting_started
- Cisco, "Cisco Intersight Workload Optimizer Target Configuration Guide," https://www.cisco.com/c/dam/en/us/td/docs/unified_computing/ucs/Intersight/Intersight_Workload_Optimizer/Cisco_Intersight_Workload_Optimizer_Target_Configuration_Guide.pdf
- Cisco, "Cisco Intersight Workload Optimizer User Guide," https://www.cisco.com/c/dam/en/us/td/docs/unified_computing/ucs/Intersight/Intersight_Workload_Optimizer/Cisco_Intersight_Workload_Optimizer_User_Guide.pdf

This page intentionally left blank

Index

A

D

I

K

P

R

S

T

U

V

W