ıı|ııı|ıı cısco

# Cisco ThousandEyes

Digital Experience Monitoring and Troubleshooting



ciscopress.com

AARON TROMPETER ROBERT WEBB



# **Cisco ThousandEyes**

# **Digital Experience Monitoring and Troubleshooting**

Aaron Trompeter Robert Webb

**Cisco Press** 

### **Cisco ThousandEyes: Digital Experience Monitoring and Troubleshooting**

Aaron Trompeter, Robert Webb

Copyright © 2025 Cisco Systems, Inc.

Published by: Cisco Press Hoboken, New Jersey

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

### \$PrintCode

Library of Congress Control Number: 2024942159

ISBN-13: 978-0-13-830918-3 ISBN-10: 0-13-830918-3

### Warning and Disclaimer

This book is designed to provide information about using Cisco's ThousandEyes platform to monitor networks, applications, and dependency services that impact user experience. 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**

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

### **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.

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

GM K12, Early Career and Professional Learning:	Copy Editor: Bill McManus
Soo Kang	Technical Editors: Jeffrey Drury, Collin Sullivan
Alliances Manager, Cisco Press: Caroline Antonio	Editorial Assistant: Cindy Teeters
Director, ITP Product Management: Brett Bartow	Cover Designer: Chuti Prasertsith
Executive Editor: Nancy Davis	Composition: codeMantra
Managing Editor: Sandra Schroeder	Indexer: Timothy Wright
Development Editor: Ellie C. Bru	Droofroador Daybara Maak
Senior Project Editor: Tonya Simpson	Prooffeader: 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.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

### About the Authors

Aaron Trompeter is a technical solutions architect within the ThousandEyes global enterprise segment at Cisco, focusing on visibility and operational awareness for onprem, SaaS, and cloud native. In this role, he aligns his passion for education and learning with his motivation for helping the infrastructure community grow and learn to harness tools to provide use cases that fit each organization. Prior to this role, Aaron spent 6 years as a data center TSA within Cisco and had a few other roles within Cisco as a service provider specialist and software engineer in the Cloud Engineering unit. Aaron has more than 20 years of experience in the IT and engineering areas and has continued to focus on networking and software.

**Rob Webb** began his technical career when, at 17 years old, he enlisted in the military as a teletype technician. His military service spanned more than 28 years, during which time he trained and worked in telecommunications systems, F-4 and F-16 avionics systems, and, eventually, Cyber Warfare Operations. Most of this time was spent working part-time in the Air National Guard. This allowed Rob to not only get the benefits of military training and experience but also to pursue a full-time career outside the military. He went from pulling cable to IT professional services to protocol analysis to pre-sales engineering/solutions architecture. Rob even ran his own consulting service, WEBNET Communication, where he achieved industry-recognized certifications in protocol analysis, which include Certified Network Expert (CNX), Sniffer Certified (SCM), and Certified NetAnalyst.

Rob has spent his career solving complex problems for customers and helping them design and implement monitoring solutions so they can better solve those problems themselves. He has spent many hours collecting and analyzing packets to understand not only what is happening in each situation, but also what should be happening and what can be done to improve it. He often refers to these as opportunities for optimization. Rob joined ThousandEyes as a sales engineer in March, 2020.

When Rob is not working, he keeps busy by enjoying outdoor activities such as fishing and photography. Now he combines the two in his sportfishing photography. You can follow Rob on LinkedIn (www.linkedin.com/in/robertwebbcyberguy) or Instagram (@photographyinspiredbylife).

### About the Technical Reviewers

Jeff Drury is a passionate technologist and self-proclaimed nerd. While at Cisco, Jeff has worked with large enterprise customers to build and enhance their data center and observability architectures. Jeff has also been an advocate of automation and programmability by developing and leading several internal and external training sessions. Prior to joining Cisco, Jeff was Director of Engineering at a regional VAR, where he built several teams and mentored early-in-career engineers. Jeff is an avid cyclist who loves to go on adventures with his wife and two teenage sons.

**Collin Sullivan** works as a solutions engineer at ThousandEyes and has specialized in helping enterprise customers design and implement infrastructure and hybrid-cloud monitoring strategies. Collin has worked with customers across verticals such as financial, manufacturing, healthcare, and technology. Before ThousandEyes, Collin worked as a consultant for Cisco, focused on the industrial wireless space, but started his journey at Cisco working in Advanced Lab Operations as part of a team that rapidly rebuilt customers' networks in Cisco's mock data centers when outages or issues occurred to determine root cause. In that sense, Collin's whole career has been focused on network issues and helping customers avoid and identify bottlenecks. When you can't find Collin reading or writing about networks, he is typically enjoying a cold beverage on the beach.

### **Dedications**

### Aaron Trompeter

I would like to dedicate this book to my family: my wife, Sarah, encouraged me to keep pushing and never procrastinate; my son, Aiden, cheered me on when writing the book through the weekends and at night; and my mother and sister said, "just do it," and, "if you want it, go get it." I could not have even started this book without the support of my family.

### Rob Webb

I would like to dedicate this book to my sister, Sue, who has always been my inspiration to do more, learn more, and pay better attention to the details. To my wonderful daughter, Lauren, who is sick of hearing me talk about this book but puts up with me all the same. And to my customers over the years for trusting me to help solve some really challenging issues we have run into. Without you, I would never have had the stories to share or the experience to sound like I know what I am talking about. The best advice I have for all of you is this: when you get stuck on a problem, call your mom. Not to get technical advice, but to take a break from the technical for a few minutes. Also, she'll appreciate it.

### Acknowledgments

### Aaron's Acknowledgments

Rob Webb, thank you for saying, "this sounds like a great idea, it will be fun." Barry Wayne, thanks for recommending Rob. Collin Sullivan and Jeff Drury, we could not have done this without you guys. Jason Warfield and Bill Donoghue, you guys allowed me onto the ThousandEyes team and have always encouraged the team to push and promote the platform.

### Rob's Acknowledgments

I'd like to thank Bill Donoghue, who trusted me when I wanted to do things a little different from what is "normal" in our field. Bill helped me to start telling my stories over a decade ago. Also, I thank Bernie Clairmont for continually setting aside time for us to brainstorm ideas and work through issues that I may be struggling with. It was Bernie who provided the framework for the Output/Input/Processes methodology (see Chapter 7). Thanks to Jim Marcel, Greg Mathews, and Joe Clark, who helped me learn many lessons as they trusted me to restore or improve services to their organization more times that I can count...thank you for helping me build stories worth sharing. Finally, thank you, Aaron, for inviting me to join in the writing of this book and giving me a place to share my stories and experience.

### **Contents at a Glance**

Introduction xvi

- Chapter 1 Introduction to ThousandEyes and Its Capabilities 1
- Chapter 2 Agent Setup 17
- Chapter 3 Configuring Tests 49
- Chapter 4 Configuring Alerts 101
- Chapter 5 Dashboards 111
- Chapter 6 Monitoring and Troubleshooting Network Performance Issues 123
- Chapter 7 Scripted Synthetic User Testing with Transaction Tests 147
- Chapter 8 Integrations 163
- Chapter 9 Best Practices: Test Optimization, Collaboration, and Stories from the Field 203
- Chapter 10 Device Monitoring 255
- Chapter 11 Account Administration 275
- Chapter 12 Automation—Use Cases and Case Studies 295
- Chapter 13 Business Strategy for Engineers 309
- Appendix A: Answers to Review Questions 315
- Appendix B: Troubleshooting 323

Glossary 333

Index 337

### **Reader Services**

**Register your copy** at www.ciscopress.com/title/9780138309183 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<sup>\*</sup>. Enter the product ISBN 9780138309183 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

Introduction xvi

Chapter 1	Introduction to ThousandEyes and Its Capabilities 1
	How ThousandEyes Was Conceived 1
	Understanding ThousandEyes Use Cases 2
	Fundamental Network Metrics 3
	It's the Network 3
	Operational Impact 3
	Packet Loss 4
	Network Latency 7
	Maximum Theoretical Throughput 11
	Results 15
	Review Questions 16
Chapter 2	Agent Setup 17
	Enterprise Agents 17
	Enterprise Agent Installation 18
	Cisco Application Framework Installation 18
	Open Virtual Appliance Installation 25
	Enterprise Agent Using Docker 33
	Enterprise Agent Using Linux 37
	Enterprise Agent Utilization 39
	Agent Cluster and Agent Clustering 41
	Enterprise Agent Placement 44
	Cloud Agents 45
	Endpoint Agents 46
	Review Questions 48
Chapter 3	Configuring Tests 49
	Tests for Enterprise and Cloud Agents 49
	Routing Layer 49
	Configure Inside-Out BGP Visibility 51
	Network Layer 53
	Agent to Agent Test 53
	Agent to Server Test 56
	DNS Layer 58
	DNS Server Test 59

DNS Trace Test 62 DNSSEC Test 64 Web Layer 66 HTTP Server Test 66 Page Load Tests 70 Transaction Test 73 FTP Server Test 74 Voice Layer 76 SIP Server Test 78 RTP Stream Test 80 Endpoint Agent Tests 82 Adding Multiple Tests to a Template 90 Dynamic Tests 90 Real User Tests 97 Monitored Domain Set 97 How Real User Tests Work 97 Review Questions 99

### Chapter 4 Configuring Alerts 101

Alert Rules 101 Alert Conditions 103 Alert Notifications 104 Email Notifications 104 Webhooks 105 Alert Suppression 107 Finding the Source of the Alert 109 Alert History 110 Review Questions 110

### Chapter 5 Dashboards 111

Dashboard Basics 111 Widgets 114 Creating a New Dashboard 116 Review Questions 121

### Chapter 6 Monitoring and Troubleshooting Network Performance Issues 123 Understanding Views 123

Understanding Network Monitoring 125 System/Application Testing 127

	End-to-End Network Testing 127
	Hop-by-Hop Network Testing (Path Visualization) 129
	SD-WAN Performance Monitoring 134
	Network Overlay 134
	Network Underlay 135
	MPLS Performance Monitoring 136
	MTU Best Practices 137
	Overlay Network MTU 138
	Underlay Network MTU 140
	Alerting 141
	Overloaded: An NFL Story 142
	Review Questions 145
Chapter 7	Scripted Synthetic User Testing with Transaction Tests 147
	What Is Selenium? 148
	Page Load Test Versus Transaction Test 148
	Set Up and Create a Transaction Test with the ThousandEves Recorder 149
	Troubleshooting and Debugging Transaction Test Scripts 153
	Transaction Test Script Best Practices 154
	Viewing the Transaction Test Data 155
	Full Stack Observability for the Network Engineer 158
	Output 158
	Input 159
	Processes 161
	Review Questions 162
Chapter 8	Integrations 163
	API Basics 163
	Custom Webhooks 166
	ServiceNow Integration 168
	Cisco Integrations 175
	Cisco Catalyst Center (Formerly Cisco DNA Center) 175
	Cisco Meraki 181
	Cisco Webex Control Hub 185
	Cisco AppDynamics 187
	Cisco SD-WAN 192
	OpenTelemetry Integration 197
	Review Questions 202

# Chapter 9 Best Practices: Test Optimization, Collaboration, and Stories from the Field 203

ThousandEyes Test Optimization 203 End-to-End Testing: Probing Mode 206 Cumulative Acknowledgements 206 Selective Acknowledgements 208 SYN 210 Path Visualization 211 Path Trace Mode 211 No. of Path Traces 215 Probing and Path Trace Modes for Different Agent Types 216 Monitoring Collaboration Meetings 216 Webex 218 Cloud & Enterprise Agents 218 Endpoint Agents 218 Internet Insights 221 Teams 222 Cloud and Enterprise Agents 222 Endpoint Agents 227 Internet Insights 230 Zoom 231 Cloud and Enterprise Agents 232 Endpoint Agents 232 Internet Insights 233 Stories from the Field 235 FTP Throughput Analysis and Optimization 235 User Experience Versus Application Performance Metric-Call Center SLA 239 Connections Failing at Load Balancer 241 Load Balancing with 20% Failure Rate 243 When MSS Matters 244 Bizarre Failure Is Not Adding Up 248 Problem Description 248 First Progress 249 Hypothesis 250

First Significant Finding 250 Eureka! 251 Review Questions 253

#### Chapter 10 Device Monitoring 255

ThousandEyes Device Monitoring 255 Device Discovery 259 Scheduled Discovery 262 Discovered Devices 264 Device Details 265 Device Alerts 266 Device Notifications 269 Device Views 269 Path Visualization with Device Monitoring 272 Operational Insights 273 Replacing a Device with a New Piece of Hardware 273 Polling 273 Topology 273 Agent Clusters 274

### Chapter 11 Account Administration 275

Review Questions 274

Adding a New User 275 Default Roles 276 Account Groups 277 Role-Based Access Control 279 Single Sign-On with ThousandEyes 281 Identity Provider Setup 282 ThousandEyes Setup 282 Imported Metadata Configuration 284 Activity Log 285 Time Selector 286 Add a Filter 287 Download Logs 288 Usage 289 Usage Tab 290 Quotas Tab 291 Review Questions 293

### Automation—Use Cases and Case Studies 295 Chapter 12 Use Case 1: Automated Testing Basics 295 Webhooks on GitHub 296 Use Case 2: Jenkins 297 Use Case 3: Ansible Playbook 299 Use Case 4: Kubernetes 300 Scaling ThousandEyes via Automation 301 Operationalizing 301 Dashboards for Comparisons 301 Inside Your Network vs. Outside Your Network (the Internet) 302 Comparing ISPs 304 Comparing In-Office and Outside-of-Office User Experience 304 Comparing Inside a Building by Floors and Zones 306 Data Center Monitoring 306 VXLAN Versus Traditional Layer 2/Layer 3 307 Agent Placement 307 Additional Note for Cisco Nexus Devices 308 Review Questions 308 Chapter 13 **Business Strategy for Engineers** 309

- Bridge the Divide: Aligning Technical Expertise with Business Strategy 309
- Appendix A: Answers to Review Questions 315
- Appendix B: Troubleshooting 323

Glossary 333

Index 337

### **Command Syntax Conventions**

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- Boldface indicates commands and keywords that are entered literally as shown.
   In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a show command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (I) separate alternative, mutually exclusive elements.
- Square brackets ([]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.

### Introduction

You might have heard Cisco ThousandEyes referred to as the "Google Maps for the Internet," the Internet visibility platform that the world's biggest brands and fastest growing startups rely on to ensure they're delivering the best possible digital experiences. This comparison isn't far from the truth. In this book we review some basic network concepts before jumping into the installation of agents. We explore where to deploy the agents and why. We look at test creation, when to use certain tests, and the objective of those tests. As we build on each chapter, you will learn how to look at alerts and what to do with them. Next, we look at operationalizing ThousandEyes through integration with other Cisco platforms and third-party apps. The goal of the book is to get you up to speed with the ThousandEyes platform quickly so that you can start creating value for your business.

### Goals, Objectives, and Approach of the Book

This book was written with the intent to help IT professionals better understand how to use Cisco ThousandEyes. We cover many how-to topics, some of which are not always obvious in the product documentation available online. Additionally, by describing realworld experiences where ThousandEyes helped to solve complex problems, we hope to help you consider additional use cases and ways to leverage the information provided by the ThousandEyes platform.

Before you begin reading the hands-on chapters, we highly recommend that you create your own free 15-day trial account, which will allow you to follow along with recommended configurations, tips, and best practices throughout the book. The link www.thousandeyes.com/booktrial provides an easy way to start your new trial account. This account will provide you with access to ThousandEyes resources such as Internet Insight and many of the Cloud Agents and enable you to install your own Enterprise and Endpoint Agents.

This book is our attempt to truly teach you how to think about ThousandEyes holistically as opposed to simply telling you what to do.

You should understand that due to the dynamic nature of Software as a Service (SaaS) hosting combined with the fast-paced innovation in which Cisco ThousandEyes operates, many changes are occurring weekly to this product. You can view and keep track of these changes, along with the dates they were implemented, in the ThousandEyes Changelog (https://docs.thousandeyes.com/whats-new/changelog). Over the course of writing this book, several features and product names have changed in Cisco ThousandEyes. For example, with Endpoint Agents, Automated Session Tests (ASTs) are now referred to as Dynamic Tests. Depending on when a particular chapter was written, you might see either of these names used. Additionally, the version of Views has recently changed from Views 1.0 to Views 2.0 for most Cloud Agent and Enterprise Agent tests. Endpoint Agent views will likely have changed by the time this book is published.

While some of the stories in this book are taken directly from our experience in using ThousandEyes to troubleshoot and solve a given problem, other stories describe how a problem was analyzed and resolved using traditional methods, followed by analysis of how the same problem could be solved much faster by using ThousandEyes. What used to take hours, sometimes days, to collect and analyze can now often be done in seconds, minimizing the requirement for extensive packet collections.

### Who Should Read This Book?

While the obvious audience would be anyone who works with ThousandEyes in their jobs, this book has been written with a far broader audience in mind: network administrators, DevOps engineers, and IT professionals who are responsible for managing and optimizing the performance of an organization's network infrastructure and applications.

Anyone who intends to apply for an IT position with a company that uses ThousandEyes should absolutely read this book. For that matter, anyone who is looking to learn about new tools and troubleshooting methodologies for network and application performance would benefit from reading this material. Because this book includes the ability to create your very own ThousandEyes demo account, it opens the field to anyone who is interested in modern IT operations and digital experience monitoring.

### How This Book Is Organized

The book is organized into 13 chapters:

- Chapter 1, "Introduction to ThousandEyes and Its Capabilities": This chapter describes the origins of ThousandEyes and introduces some of its key use cases. It also addresses fundamental networking concepts that you need to maximize your understanding of the technologies, tests, and metrics discussed throughout the book.
- Chapter 2, "Agent Setup": This chapter explains agents in ThousandEyes, including Cloud (Public) Agents, Enterprise (Private) Agents, and Endpoint (User) Agents, which are all used to initiate tests.
- Chapter 3, "Configuring Tests": This chapter addresses the different ThousandEyes tests that can be deployed as well as the parameters that are generally used for each test type.
- Chapter 4, "Configuring Alerts": Once you understand the tests and test types, it's time to learn about how ThousandEyes alerts when thresholds are exceeded or certain conditions are or are not met.
- Chapter 5, "Dashboards": This chapter discusses how ThousandEyes dashboards allow metrics to be viewed across tests, agents, locations, and more.
- Chapter 6, "Monitoring and Troubleshooting Network Performance Issues": This chapter covers the different tests used to provide both end-to-end metrics and hop-by-hop metrics. Understanding how these tests are executed in detail will help you to troubleshoot issues when they arise.

- Chapter 7, "Scripted Synthetic User Testing with Transaction Tests": This chapter explains how ThousandEyes enables customers to record multistep, transaction test scripts and then deploy them as a synthetic test. These Transaction tests exercise the target application by accessing multiple pages and even authenticating (where applicable).
- Chapter 8, "Integrations": ThousandEyes has the capability to integrate not only with other Cisco products, such as Application Dynamics (AppD), but also any applications using webhooks. Recently, ThousandEyes has added the capability to export OTel data from the API in addition to the traditional query method. This chapter discusses these various integrations and how they are beneficial.
- Chapter 9, "Best Practices: Test Optimization, Collaboration, and Stories from the Field": Because all networks are different, sometimes tests need to be "tweaked" to achieve the best results. This chapter covers many of these scenarios and discusses how and when to tweak.
- Chapter 10, "Device Monitoring": This chapter explains how ThousandEyes monitors devices via SNMP. This enhances the traditional Path Visualization described throughout the book with device-level metrics such as CPU, Memory, Error, and Discards.
- Chapter 11, "Account Administration": This chapter gets into the weeds of account administration, including adding users and defining their roles within ThousandEyes, which is critical to operating a comprehensive and secure SaaS platform.
- Chapter 12, "Automation—Use Cases and Case Studies": This chapter relates some of the field experiences of the authors, providing insight into the application of ThousandEyes. The ability to automate the deployment of agents and tests is critical to some of our largest customers, and this chapter provides some examples of this.
- Chapter 13, "Business Strategy for Engineers": This chapter discusses how ThousandEyes plays a part in your company's business strategy and what that means to a network engineer.

### **Figure Credits**

Figures 8-7 through 8-23 © 2024 ServiceNow

Figure 9-43 inset courtesy of Defense Logistics Agency

# Chapter 5

# Dashboards

When you log in to the ThousandEyes platform, the first view that you see is the Dashboards page, containing charts and easy-to-read graphs summarizing your data. It provides dynamic and real-time insights into your network and application performance. This chapter guides you through the intricacies of ThousandEyes dashboards, helping you to harness the full potential of this essential feature.

As you enter the Dashboards page, you are greeted with live status dashboards, offering a snapshot of a specific time period (by default, 24 hours). Additionally, you have the capability to schedule and share point-in-time snapshots of a dashboard. You can organize the information derived from tests and Internet Insights into highly customized layouts. These layouts can be presented numerically, in tables, or through intuitive graphs, allowing for a comprehensive understanding of your network and application health.

A noteworthy feature of the Dashboards page is that it automatically refreshes every 2 minutes, ensuring you are consistently updated with the latest data. This not only facilitates active monitoring but also prevents automatic logouts due to inactivity. This attribute makes the Dashboards page suitable for deployment in kiosks or operations center displays.

This chapter introduces the various components, customization options, and strategies for effective utilization of ThousandEyes dashboards.

### **Dashboard Basics**

When you log in to the ThousandEyes UI the Dashboard menu is the first page you see. By default, the dashboard uses the test data that has already been created to populate its visualizations.

As an example, Figures 5-1 and 5-2 show the defaults where the test data is represented in a basic dashboard view; we will look at how to customize for more granular specifics.

三 🖞 Thousand Eye	s Dashboards			🝞 🚍 🎫 🌲 🚥 🛛 🖓 🗛 Aaron Trompeter Demo - Aaron Trompeter ~
Cloud & Enterprise Agents >	Dashboard: ThousandEyes Built-in 💌			Expand [ Last 24 Hours • () C 2m • () ± + New
Endpoint Agents >	Tests Cloud & Enterprise Agents 12 hours			
Devices >	Test Name	Test Type	Alert Status 👃	Trending (12h) / Current Values
Internet Insights >	VPN2-PACCAR	Web - HTTP Server	2 alerts	
Dashboards	парылоло, гост			N/A
Alerts 6 >				MA
Integrations	https://espn.cpm	Web - Page Load	2 alerts	
Sharing >				1 - 1
Account Settings >	VPN3-PACCAR 8.40.118.1	Network - Agent to Server	1 alert	18.4% 36.13 ms
	VPN-1PACCAR 70.183.74.86	Network - Agent to Server	1 alert	100% NA
	MGM Edge 69.162.16.9	Network - Agent to Server	1 alert	100% NVA
I wish this page would	www.churcholjesuschrist.org:80 www.churcholjesuschrist.org	Network - Agent to Server	•	

Figure 5-1 Dashboard Defaults

= the Thousand Eyes	s Dashboards		0 🗎	93• 🙏 99• 🕴 🚺	Aaron Trompeter Demo - Aaron Trompeter
Cloud & Enterprise Agents >	Dashboard: ThousandEyes Built-in V Filters		Last 24 Hours 👻 🎯	●   2 2m •   G	🗄 📥 🕂 New \cdots
Endpoint Agents >	Alert List ( Alerts) Widget Filters — 1 hour				
Devices >	Total Alerts 19 . Active Alerts 8				View Active Alerts
Internet Insights >					
Dashboards	↓ Alert Source	Alert Rule	Alert Type	Start Time (UTC)	Duration
Alerts 🙃 >	VPN-1PACCAR	Default Network Alert Rule 2.0	Network - Agent to Server	3 months ago	100d 1h 57m 44s
Integrations	VPN2-PACCAR	Default HTTP Alert Rule 2.0	Web - HTTP Server	3 months ago	100d 1h 57m 44s
Sharing >	• espn	Default Page Load Alert Rule	Web - Page Load	7 days ago	6d 23h 3m 44s
Account Settings >	google.com ICMP	Aaron-Test-PathTrace	Network - Path Trace	2 minutes ago	1m 44s
	VPN2-PACCAR	Default Network Alert Rule 2.0	Network - Agent to Server	3 months ago	100d 1h 57m 44s
	VPN3-PACCAR	Default Network Alert Rule 2.0	Network - Agent to Server	20 minutes ago	19m 44s
	• espn	Default HTTP Alert Rule	Web - HTTP Server	7 days ago	6d 23h 3m 44s
	MGM Edge	Default Network Alert Rule	Network - Agent to Server	13 days ago	13d 0h 57m 44s

Figure 5-2 Dashboard Defaults Continued

When the tests shown in Figure 5-1 trigger alerts, the details of the alerts are displayed in the Alert List widget shown in Figure 5-2, including the alert rule that triggered the alert and the alert type.

You can quickly customize your Dashboards page by adding one or more of the built-in dashboards that ThousandEyes provides. From the Dashboard drop-down list, choose **ThousandEyes Built-in**, as shown in Figure 5-3, to see other built-in dashboards that might fit your needs. Notice that the dashboard categorizes the tests, so creating structure early on will help users operationalize and locate metrics more easily.

	_		
Cloud & Enterprise Agents	>	Dashboard: ThousandEyes Built-in 🔻	<b>V</b> Filter
Endpoint Agents	>	Q. Filter by dashboard name	
		Selected Dashboards	
Devices	>	ThousandEyes Built-in	国方
Internet Insights	>		
		Dashboards 11 Dashboards	
Dashboards		Internet Insights Built-in	臣方
Alerts	>	ThousandEyes Built-in: Agent Alerts	r
Integrations		ThousandEyes Built-in: HTTP Server	ið "
Sharing	>	ThousandEyes Built-in: Network Layer	t i
• • • • •	ĺ.	ThousandEyes Built-in: Page Load	i i

Figure 5-3 Dashboard Built-ins

As an example, Figure 5-4 shows the HTTP Server built-in dashboard. You can see a completely different experience versus the other default dashboard. The point of this is to show examples of what metrics can be displayed and how layouts can be presented.

≡ clisco ThousandEye	sco   Dashboards	😧 🚍 🚥 🌲 🚥 🗌 🐼 Aaron Trompeter	
Cloud & Enterprise Agents >	Dashboard: ThousandEyes Built-in: HTTP Server 🔻	Last 24 Hours ▼ (©)	
Endpoint Agents > Devices > Internet Insights > Dashboards Alers (1) >	Average Availability         Average Response Time         Average           85.008%/Maan         193.7 ms <sup>-1,410ms</sup> 200.           Web - HTTP Bover - Negative File         21 table - A4 doese - Negative File         200.	© Throughput 95Mbps/Maan Parent Parent	
Integrations Sharing > Account Settings >	Overall Average Availability ( Over & Emergen Agent) Web - HTTP Barver — Availability - Weiger Filters — 1 hour	— Mean of Availability	
	0	1616 1530	
I wish this page would ThousandEyes © 2024 Privacy	Overall Average Availability ( Ocud & Enterprise Agents) Web - HTTP Server — Availability - Widget Filters — 1 hour	Average Availability - Bottom 10 Tests (C Goud & Enterprise Agents) Web - HTTP Server — Availability - Widget Filters — 1 hour	

Figure 5-4 HTTP Server Built-in Dashboard

To help you maneuver around the dashboard, Figure 5-5 provides a quick reference for the various functions in the dashboard.

The Add Widget button and ellipsis menu shown in Figure 5-5 are the typical starting points, respectively, for adding widgets to a dashboard and creating a new dashboard, or editing or deleting an existing dashboard. These controls enable administrators to organize specific data and structure it according to a particular use case. For example, grouping Agent to Agent tests that connect to the data center enables administrators to swiftly identify issues and generate high-level reports.



Figure 5-5 Dashboard Built-in Controls

### Widgets

In the context of ThousandEyes dashboards, *widgets* are visual elements that represent specific types of data or metrics. Widgets can represent data from alerts, agents, and tests. They provide a dynamic and interactive way to present information relevant to your monitoring and testing needs. The following are some common types of widgets that you might encounter when expanding or customizing a dashboard:

- Widget Type: Live Status
  - Agent Status: Offers a live look at the status of your Enterprise or Endpoint Agents to give you an idea of overall agent health.
  - Tests: A 12-hour, live display of a list of tests configured in your account group for a one-stop glance at high-level test health. Grayed-out rows show disabled tests.
  - Alert List: Provides a look at the alerts that were active during the configured time interval.
- Widget Type: Breakdown
  - Stacked Bar: Provides horizontal histogram bars with multiple values, useful for composite metric data and for comparing values between multiple tests or on a per-country basis.
  - **Grouped Bar:** Represents multiple values as single bars in a group of bars, oriented either horizontally or vertically.

- Pie: Similar to Stacked Bar chart widgets, representing data in a circular statistical graphic. It is used to illustrate numerical proportions, which makes it easier to understand the distribution of the selected metric.
- Widget Type: Data Summary
  - **Table:** Allows a breakdown of numbers by rows and columns, listing by test, country, continent, or data source.
  - Multi-Metric Table: Can have columns with different metrics, displaying more varied information.
  - Number: Displays one or more cards, each showing a single scalar quantity or a number of alerts.
  - Color Grid: Displays an array of colored cards, where each card's color depends on the configured color scale.
- Widget Type: Time Series
  - Line: This widget displays data using a line plot, where time is represented on the horizontal axis and the selected metric or quantity is shown on the vertical axis. This type of visualization is particularly useful for observing trends and patterns over a specified time period.
  - Stacked Area: Line plots showing quantities over time, similar to stacked bar charts but representing values over time.
  - Box and Whiskers: Plots data values versus time on the horizontal axis, with the vertical axis displaying the median, minimum, and maximum data points per time value.
- Widget Type: Maps
  - Map: Displays data on a world map based on the location of testing systems, with options to show data per country, continent, or per agent. This allows the user to quickly determine whether a specific region is experiencing issues.

These widgets offer a versatile set of tools to visualize and analyze data from different perspectives, catering to various monitoring needs. To access the widgets described in the previous list, from the dashboard, click **Add Widget** to see all the widgets (see Figure 5-6).

Adding widgets to a built-in dashboard is a common practice to tailor the display according to specific needs. The purpose of each widget is to provide a clear visualization of different data metrics. In the next section, we delve into the process of creating a new dashboard, offering you the flexibility to curate your own visualizations based on the available widgets.

Add a Widget		^	×
LIVE STATUS			
	• • • •		
Agent Status	Tests	Alert List	
BREAKDOWN			
Stacked Bar	Grouped Bar	Pie	
DATA SUMMARY			
		1 2 3	
Table	Multi-metric Table	Number	
Color Grid			

Figure 5-6 Widgets

### **Creating a New Dashboard**

Creating a dashboard in ThousandEyes is as straightforward as adding a widget to a builtin dashboard. To effectively use dashboards, it's essential to understand the different widget types and the data they display. This ensures that the relevant metrics are visualized in a way that best supports your monitoring needs.

From the dashboard, click the **New** button and choose **Dashboard** (see Figure 5-7) to create a new dashboard and add widgets to it.

≡ the state = = = = = = = = = = = = = = = = = = =	es 💿 🛛 Dashboards			© ?	<b></b>	<u> .</u> 🚥	<li>Aa De</li>	aron Trompeter emo - Aaron Trompet	ter ~
Cloud & Enterprise Agents >	Dashboard: ThousandEyes Built-	in: Network L 🔻 🍸 Filters		Last 24 Hours 👻	0	Ĵ 2m +	Ø	는 + New	
Endpoint Agents >								Widget	
	Network Layer Overview							Dashboard	
Devices >	Average Loss	Average Latency	Average Jitter						
Internet Insights >									
	3.99% <sup>▲0.02%</sup>	27.08ms <sup>v0.02ms</sup>	1.44ms <sup>A0.12ms</sup>						
Dashboards	Network - Agent to Server Pack	Network - Agent to Server - Late	Network - Agent to Server - litter						

Figure 5-7 Create New Dashboard Menu Selection

In the Create New Dashboard dialog box that opens, shown in Figure 5-8, give the dashboard an intuitive name and choose which account group(s) will be able to view the dashboard. In the View Settings, you can check **Set As Private** if you don't want anyone else to be able to view the dashboard, or you can check **Set As My Default** or **Set As Default for Account Group** if you want this dashboard to be displayed by default on your own Dashboards page or the account group's Dashboards page, respectively. Click **Create Dashboard** to create the shell dashboard and return to the Dashboards page to configure it with widgets.

	Create New Dashboard ×
	Name
	Enter a Name
	Labels Select Label(s)  Manage Labels
	Description
Enter a Descriptive Dashboard Name	Enter Description
	Default Settings
	Time Range
	Last 24 hours
	Auto Refresh Rate
	Select an item *
	Visibility
Set the Account	Account Group Visibility
View Settings	Only current account group ~
view octariys	Cancel Create Dashboard

Figure 5-8 Dashboard Creation

On the Dashboards page, locate your new dashboard and click its **Add Widget** button to open the Add a Widget panel (refer to Figure 5-6). Click a widget to open a separate panel in which to configure it. As an example, Figure 5-9 shows the panel that opens when adding the Color Grid widget (Data Summary type). When creating a new widget, all the options can seem overwhelming. Focus on how you want the data to look, and experiment with the various widgets to find what suits you (click Cancel instead of Save to return to the Add a Widget panel and choose a different widget to view).

**Step 1.** Data Source: Select a data source from the Data Source drop-down menu. Most users may use the default option, Cloud & Enterprise Agents, but it's crucial to consider other options, such as Internet Insights or Endpoint Agents.

Editing Color Grid	^ X
Color Grid	
Data Source	
Cloud & Enterprise Agents	•
Category	
Select a Category	•
Metric	
Select a Category first	•
Measure	
Select a Metric first	•
Scale auto to auto	
DESIGN	
Cards	
Select a Property -	
Group Cards By	
Select a Property -	
Sort Cards By	
Alphabetical	
Save & Add Another Widget	Cancel Save

Figure 5-9 New Dashboard Widget

**Tip** Choose a data source that aligns with the specific data you wish to visualize.

**Step 2.** Category: Choose the category. Specify what aspect or metric you want to monitor from the selected data source. For example, if you have chosen Cloud & Enterprise Agents as your data source, you might select Web - HTTP Server as the category to display metrics related to HTTP server performance.

**Note** The options available in the Category drop-down list will depend on the data source you selected.

**Step 3.** Metric: Select the specific metric you want to monitor. You have several options to choose from, including Availability, DNS Time, or SSL Time. For the purposes of this example, let's select Availability as the metric.

**Note** You can select only one metric per widget. If you want to observe multiple metrics, you will need to create additional widgets for each metric.

**Step 4.** Measure: After selecting the metric, you'll need to determine how you want to measure it by choosing an option from the Measure drop-down list. The options include Median, Maximum, and Minimum.

**Tip** I recommend choosing Median because it provides a balanced view, avoiding the extremes that maximum or minimum values might present. There are other options such as nth percentile and standard deviations.

- **Step 5.** Cards: Select Tests from the Cards field drop-down menu. If you are using categories, use the Group Cards By field to organize them accordingly. This helps in sorting and grouping specific tests or agents. Although not covered in previous chapters, labeling enables users to group specific tests or agents together. For this example, choose All to include all relevant tests or agents.
- **Step 6.** Set up filters: Scroll down to the Filter By section (this section is not shown in Figure 5-9) and select Tests to filter the data. If you have numerous agents worldwide running those tests and want to focus on specific agents, add additional filters to define your data more precisely.
- Step 7. Save and view the data: Click Save to apply the data and the layout to your new widget. Figure 5-10 shows the example Color Grid widget. There are additional widgets you can include: Add Agent Status and Endpoint Agents represent Real-Time Agent Status and End User Experience status in the dashboard (see Figure 5-11).



Creating a dashboard allows your imagination to come alive and paint the picture of the network from many perspectives. Dashboards are fully customizable and can be created for a single use case if needed.

Another great feature of widgets is that you can duplicate a widget, change a metric or two (for example), and save the duplicate as a new widget. That way, you don't have to go through the entire process of creating a new widget every time you need one.



Figure 5-11 Agent Widget Example

The Embed Widget feature, shown in Figures 5-12 and 5-13, allows you to grab the data and use it as an iframe to insert in a web page, wiki page, or anything external that can use an iframe.

Network Layer Overview
Average Loss Average Latency Average Jitter
3.32 <sup>40%</sup> 53.51 <sup>ms</sup> <sup>V0.17ms</sup> 0.88 <sup>ms</sup> Mean
Network - Agent to Server — Pack Network - Agent to Server — Late Network - Agent to Server — Jitter 28 Tosts - All Ansatz - 1 day.

Figure 5-12 Widget Features



Figure 5-13 Iframe Widget

Returning to the dashboard widgets, if you encounter errors or issues, you can click on the widget to identify which test(s) are experiencing problems (see Figure 5-14). When you click on the widget, a window will appear to show all tests that are associated within that widget. In the next chapter, we explore how to monitor and troubleshoot network performance issues.

Network Layer Overview				
Average Loss	Average Latency	Average Latency		~ ×
<b>3.32%</b> <sup>A0%</sup> Network - Agent to Server — Pack	<b>53.51 ms</b> Mean Network - Agent to Server — Late			16 Tests
26 Tests • All Agents • 1 day	26 Tests • All Agents • 1 day		Test Name	Latency 🗸
Overall Average Loss ( Cloud & Enterprise Agents) Network - Agent to Server — Packet Loss + 1 day			Nyopi uchadh org HD	161.67 ms
			Transmittym Search	132.74 ms
%			Rever Salation	73.60 ms
15	٨		Tarastin, Ann	52.87 ms
10			Property land-040	51.88 ms
10			MOR Two CMP	21.90 ms
5				

Figure 5-14 Associated Test to Widget

### **Review Questions**

Answer the following questions. Check your answers against those provided in Appendix A, "Answers to Review Questions."

- 1. Can you create a widget that shows only alerts from Internet Insights?
- 2. Is there a way to extract the dashboard data to a web page?
- **3.** When creating a widget, can you add more than one metric? If not, how can this be done?
- **4.** What would be the best widget to use when looking at end users' wireless signal strength?

This page intentionally left blank

# Index

# **Numerics**

3-way handshake, TCP, 7, 207, 236

# Α

account administration account groups, 277-279 Activity Log, 285-286 add a filter, 287-288 download logs, 288–289 Quotas tab, 291–293 time selector, 286–287 usage, 289-291 adding a new user, 275-276 default roles, 276-277 RBAC (role-based access control), 279-281 SSO (single sign-on), 281 *identity provider setup, 282 imported metadata* configuration, 284-285 ThousandEyes setup, 282–284 account groups, 277-279

Activity Log, 285–286 add a filter, 287–288 download logs, 288-289 Quotas tab, 291–293 time selector, 286-287 usage, 289-291 Agent to Agent test, 53–56 Agent to Server test, 56–58 Agents Cisco Device, 44 Cloud, 45–46 Cluster, 39-40, 41-43, 274 Docker container, 44 Endpoint, 46-48 Enterprise, 17–18, 159–160 Docker container, 33–37 installation, 18-24, 25-33 Linux, 37-39 placement, 44-45 troubleshooting, 327-329 utilization. 39-41 Linux, 44 VM (Virtual Machine), 44

alert/s, 101–102 conditions, 103 device, 266-269 finding the source of, 109–110 history, 110 metrics, 103-104 MTU (Maximum Transmission Unit), 141-142 notifications, 104 email, 104 webbooks, 105-107 renaming, 104 rules, 101–103, 168 suppression, 107–109 Ansible, 299–300 API from browser, 165 data-streaming, 197 ThousandEyes, 163–164 APM (application performance monitoring), 2, 239 AppDynamics, 161 app-hosting command, 19–20 application/s dependencies, 159-160 input, troubleshooting, 159–161 outages, 222-223, 234 output, troubleshooting, 158–159 processes, troubleshooting, 161–162 ARP table, 251–252 ASW (Alert Suppression Windows), 107-109 authentication, ThousandEyes API, 163 - 164automation, 105-107, 166-167 Ansible, 299–300 Jenkins, 297–299

Python script, 165–166, 295–296 webhooks on GitHub, 296–297 options, 167–168 template, 167 testing, 167

### B

bandwidth, 3
best practices
MTU (Maximum Transmission Unit),
137–138
transaction test, 154–155
BGP, 49–51
inside-out visibility, 51–53
peering request, 52–53
route hijacking, 51
route leak, 51
bidirectional network latency, 7.
See also latency
built-in dashboards, 112–113, 115
business strategy, aligning with
technical expertise, 309–313

# С

CAF (Cisco Application Framework), Enterprise Agent installation, 18–24 CAM table, 251–252 child zone, 63 chmod command, 38 CI/CD (continuous integration/ continuous delivery), 297 CIFS (Common Internet File System), 11–12 Cisco AppDynamics integration, 187–192 Cisco Catalyst Center integration, 175 - 181Cisco Meraki integration, 181-185 Cisco SD-WAN integration, 192–196 Cisco Webex Control Hub integration, 185-187 CLI utilities, 330 Cloud Agent, 45-46. See also DNS layer; network layer; routing layer; voice layer; web layer DNS layer DNS server test, 59–62 DNS trace test, 62–64 DNSSEC test, 64-66 network layer Agent to Agent test, 53–56 Agent to Server test, 56–58 test template, 216-217 voice layer, 76-77 RTP stream test, 80-82 SIP server test, 78–80 web layer, 66 FTP server test, 74–77 HTTP server test, 66-70 Page Load test, 70–73 Transaction test, 73–74 Clusters, 39, 41-43, 274 coding, 147 collaboration services, monitoring, 3 Microsoft Teams, 222 Cloud and Enterprise Agents, 222-227 Endpoint Agents, 227–230 Webex Cloud and Enterprise Agents, 218-219 Endpoint Agents, 218–221 Zoom, 231

Cloud and Enterprise Agents, 232 Endpoint Agents, 232-234 collectors, 197-198 commands app-hosting, 19–20 chmod, 38 Curl, 164–165 docker, 34-35, 327 docker ps. 37 docker run, 36 Linux, 330 run-opts, 22 show app-hosting detail, 23–25, 327-329 conditions, alert, 103 console.log function, 153 CRC/FCS check, 5 Create New dialog box, 117 creating dashboards, 116-121 new user account, 275-276 CSMA/CA (carrier-sense multiple access with collision avoidance), 11 CSMA/CD (carrier-sense multiple access with collision detection), 11 CSP (cloud service provider), 45 CSS identifiers, 154 cumulative acknowledgements, 206-208 Curl, 164-165

# D

dashboard/s, 111 built-in, 112–113 for comparisons, 301–304 *comparing ISPs*, 304–305

physical architecture, 306 user experience, 304-305 creating, 116-121 defaults, 111-112 widgets, 114-115 associated tests, 121 features, 120 iframe, 120 data, in flight, 13 data center monitoring, 306 agent placement, 307-308 Cisco Nexus devices, 308 VXLAN versus traditional Layer 2/Layer 3, 307 data-streaming APIs, 197 DBS (Data Block Size), 12 debugging, transaction test scripts, 153 - 154default roles, 276-277 dependencies, application, 159-160 Device Layer, 255–256 device/s alerts, 266-269 details, 265-266 monitoring, 255, 257 discovering devices to be monitored, 257-261 path visualization, 272–273 prerequisites, 256-257 notifications, 269 replacing, 273 scheduled discovery, 262-264 views, 269-272 discards, 6, 143-144 inbound, 6 outbound, 7 distance delay, 9-10

DNS layer DNS server test, 59–62 DNS trace test, 62–64 DNSSEC test, 64–66 docker commands, 327 Docker container Enterprise Agent, 33–37 docker ps command, 35, 37 docker run command, 36 driver.sleep function, 154 driver.takeScreenshot() function, 154 DSCP, 131–132 dynamic tests, 90–95

### Ε

email notifications, 104 Endpoint Agent, 46–48 template, 218–219 tests, 82-90 Webex. 218–221 end-to-end network test, 127-128, 203 cumulative acknowledgements, 206 - 208SACK (Selective Acknowledgements), 208 - 210SYN, 210–211 Enterprise Agent/s, 17-18, 159-160. See also DNS layer; integrations; network layer; routing layer; voice layer; web layer DNS laver DNS server test, 59-62 DNS trace test, 62–64 DNSSEC test, 64–66 Docker container. 33–37

installation CAF-based, 18–24 **Open Virtual Appliance** installation, 25–33 Linux, 37–39, 44 network layer Agent to Agent test, 53–56 Agent to Server test, 56–58 placement, 44-45 remote troubleshooting process, 126 - 127test template, 216–217 troubleshooting, 327–329 utilization, 39–41 voice layer, 76-77 RTP stream test, 80–82 SIP server test, 78-80 web layer, 66 FTP server test, 74–77 HTTP server test, 66–70 Page Load test, 70–73 Transaction test, 73–74 errors. 5 explicit tunnel, 136 exporting, transaction tests, 151-152

# F

features, widget, 120 finding the source of alerts, 109–110 in flight, 13 flow data, 125 Forrester Consulting survey, 53 Forwarding Loss, 130 FTP server test, 74–77 throughput analysis and optimization, 235–238 functions console.log, 153 driver.sleep, 154 driver.takeScreenshot(), 154

# G-H

GitHub, webhooks, 296-297

history, alert, 110 hop-by-hop test, 129–134 Path Trace Mode, 211 *number of path traces, 215–216 probing and path trace modes for different agent types, 216* HTTP Server test, 66–70 end-to-end network, 127–128 hop-by-hop network, 129–134 system/application, 127–128

ICMP (Internet Control Message Protocol), 323 Echo Reply, 7 messages, 323-324 icons, Recorder, 153 iframe widget, 120 implicit tunnel, 136 inbound discards, 6 input, troubleshooting, 159-161 inside-out visibility, BGP, 51–53 installing CLI troubleshooting utilities on a Linux distribution, 330 **Enterprise Agent** CAF-based, 18–24 OVA-based, 25-33

integrations Cisco AppDynamics, 187–192 Cisco Catalyst Center, 175–181 Cisco Meraki, 181–185 Cisco SD-WAN, 192–196 Cisco Webex Control Hub, 185–187 OpenTelemetry, 197–202 ServiceNow, 168–175 webhooks, 166–167 options, 167-168 template, 167 testing, 167 Internet, 1 Internet Insights, 221, 233 application outages, 222–223, 231.234 network outages, 222, 230-231, 233 - 234ISP (Internet service provider), 45 comparing, 304-305 outages, 221

# J-K

Jenkins DSL (Domain-Specific Language), 297–299 jitter, 127

Kubernetes Operator, 300-301

# L

labels, 306 Lad, Mohit, 1–2 latency, 7, 131, 238 distance delay, 9–10 protocol delay, 11 queuing delay, 9 routing/switching delay, 10 serialization delay, 8–9 Linux -based agents, troubleshooting, 329–330 commands, 330 Enterprise Agent, 37–39, 44 sudo commands, 330 load balancers, 143, 241–243 long-running tests, 40 LPS application, 143

# Μ

macOS, SSH into ThousandEyes, 331-332 markers, 154 methodology, troubleshooting, 158 input, 159-161 output, 158–159 processes, 161-162 metrics, 3, 125. See also widgets alert, 103–104 Forwarding Loss, 130 latency, 7 MTT (Maximum Theoretical Throughput), 11–15 output, 158–159 packet loss, 4–5 discards, 6-7 errors. 5 Microsoft Teams, monitoring, 222 Cloud and Enterprise Agents, 222 - 227Endpoint Agents, 227–230 monitored domain set, 97 monitoring data center. 306

agent placement, 307–308 Cisco Nexus devices, 308 VXLAN versus traditional Layer 2/Layer 3, 307 device discovering devices to be monitored, 257-261, 264 path visualization, 272–273 prerequisites, 256-257 Microsoft Teams, 222 Cloud and Enterprise Agents, 222-227 Endpoint Agents, 227-230 MPLS (Multiprotocol Label Switching), 136-137 network, 125 versus troubleshooting, 125 Webex Cloud and Enterprise Agents, 218-219 Endpoint Agents, 218-221 Zoom, 231 Cloud and Enterprise Agents, 232 Endpoint Agents, 232–234 MPLS (Multiprotocol Label Switching), 131–132 explicit tunnel, 136 implicit tunnel, 136 opaque tunnel, 136–137 MSS (Maximum Segment Size), 244-248 MTT (Maximum Theoretical Throughput), 11-15, 236 MTU (Maximum Transmission Unit), 244 alerting, 141-142 best practices, 137-138

overlay network, 138–140 underlay network, 138, 140–141

# Ν

NAT (network address translation), 242 National Science Foundation, 2 network bandwidth, 3 end-to-end test, 127–128 hop-by-hop test, 129-134 latency, 7 distance delay, 9–10 protocol delay, 11 queuing delay, 9 routing/switching delay, 10 serialization delay, 8-9 load balancing, 143 metrics, 3 monitoring, 125 overlay, 134, 138-140 packet loss, 4–7 discards, 6–7 errors, 5 path issues, troubleshooting, 325-326 underlay, 135, 138, 140-141 network layer, 53 Agent to Agent test, 53–56 Agent to Server test, 56–58 notifications alert. 104 email, 104 webbooks, 105-107 device, 269

# 0

Oliveira, Ricardo, 1–2 opaque tunnel, 136–137 OpenTelemetry integration, 197–202 outages, 221, 309–310 application, 222–223, 234 network, 222, 230–231, 233–234 outbound discards, 7 output, troubleshooting, 158–159 OVA (Open Virtual Appliance), Enterprise Agent installation, 25–33 overlay network, 134, 138–140

### Ρ

packet flooding, troubleshooting, 248 - 253packet loss, 4-5, 203 discards, 6-7, 143-144 errors, 5 Page Load test, 70-73, 148-149 parent zone, 62-63 Path Trace Mode, 211, 215-216 path visualization, 129-134, 272-273, 324-325 performance FTP throughput analysis and optimization, 235-238 metrics network latency, 7–11 packet loss, 4–7 MPLS (Multiprotocol Label Switching), 136–137 MTT (Maximum Theoretical Throughput), 11–15 SD-WAN, 134

network overlay test, 134 network underlay test, 135 Ping, 7 placement, Enterprise Agent, 44 - 45playbook, Ansible, 299-300 PMTUD (Path MTU Discovery), 138 PoC (proof of concept), 301-302 polling, 273 Probing Mode, 206 Force SYN, 210-211 SACK (Selective Acknowledgements), 208 - 210process switching, 6 processes, troubleshooting, 161–162 protocol analyzers, relative SEQ, 206 - 207protocol delay, 11 Python script, 165–166 reading from CSV file, 295–296

# Q-R

queuing delay, 9 Quotas, 291–293

RBAC (role-based access control), 279–281 RCA (root cause analysis), 244 Real User tests, 97 configuring, 97–99 monitored domain set, 97 Recorder, 147 CSS identifiers, 154 icons, 153 transaction tests *configuring, 149–152* 

exporting to ThousandEyes, 151 - 152saving, 152 relative SEQ, 206-207 renaming, alerts, 104 replacing devices, 273 round-trip time, 7 routing, delay, 10 routing layer BGP, 49–51 inside-out visibility, 51-53 peering request, 52–53 route bijacking, 51 route leak, 51 RTP stream test, 80-82 rules, alert, 101-103, 168 run-opts command, 22

# S

SACK (Selective Acknowledgements), 7,208-210 screenshots, 154, 157 script adding a screenshot, 157 Jenkins DSL, 297-299 Python, 165-166, 295-296 transaction test debugging, 153-154 markers, 154 **SD-WAN** performance monitoring, 134 network overlay, 134 network underlay, 135 Selenium, 147, 148 browser automation, 148 dynamic content handling, 148

element interactions, 148 WebDriver, 148 sequence numbers, TCP, 7 serialization delay, 8-9 servers, load balancing, 143 ServiceNow integration, 168–175 show app-hosting detail command, 23 - 25SIP server test, 78–80 on-site assessment, 142–143 SLA (service-level agreement), call center, 239-240 SMB (Server Message Block), 11–12 **SNMP**, 125 SSH. 330-332 SSO (single sign-on), 281 identity provider setup, 282 imported metadata configuration, 284–285 ThousandEyes setup, 282–284 sudo command, 330 suppression, alert, 107-109 switching, delay, 10 system/application testing, 127–128

### Т

### TCP

3-way handshake, 7, 207, 236
cumulative acknowledgements, 206–208
MSS (Maximum Segment Size), 244–248
packet flooding, troubleshooting, 248–253
protocol delays, 7
SACK (Selective Acknowledgements), 7, 208–210

sequence numbers, 7 SYN, 210–211 Teams. 222 monitoring Cloud and Enterprise Agents, 222-227 Endpoint Agents, 227–230 template adding multiple tests, 90 Cloud and Enterprise Agent test, 216 - 217Endpoint Agent, 218–219 webhook. 167 test/s. See also Agents; Cloud Agent; **Enterprise Agent** DNS layer DNS server test, 59-62 DNS trace test, 62–64 DNSSEC test, 64–66 dynamic, 90–95 Endpoint Agent, 82-90 end-to-end cumulative acknowledgements, 206-208 SACK (Selective Acknowledgements), 208-210 SYN. 210-211 hop-by-hop number of path traces, 215-216 Path Trace Mode, 211 probing and path trace modes for different agent types, 216 HTTP Server end-to-end network. 127-128.203 hop-by-hop network, 129–134 system/application, 127–128

long-running, 40 monitoring, 127 network layer Agent to Agent, 53–56 Agent to Server, 56-58 operationalizing, 301 optimization, 130, 203-206 packet loss, 203 Real User. 97 configuring, 97–99 monitored domain set. 97 template, 90 transaction, 147 best practices, 154–155 exporting to ThousandEyes, 151 - 152scripts, debugging, 153–154 setting up and creating with the Recorder, 149–152 voice layer RTP stream test, 80–82 SIP server test, 78–80 web layer FTP server test, 74–77 HTTP server test, 66–70 Page Load test, 70–73 Transaction test, 73–74 widget, 121 ThousandEyes, 1-2. See also Agents; alert/s; dashboard/s acquisition, 2 API. 163 authentication, 163-164 from browser, 165 ASW (Alert Suppression Windows), 107-109 Devices View. 6

Enterprise Agent, 17–18 CAF-based installation, 18–24 Docker container, 33–37 Linux, 37-39 **Open Virtual Appliance** installation, 25-33 placement, 44–45 utilization, 39–41 Forrester Consulting statistics, 53 integrations Cisco AppDynamics, 187–192 Cisco Catalyst Center, 175–181 Cisco Meraki, 181–185 Cisco SD-WAN, 192–196 Cisco Webex Control Hub, 185-187 OpenTelemetry, 197–202 ServiceNow. 168–175 webbooks, 166–168 Internet Insights, 221, 233 application outages, 222–223, 231.234 network outages, 222, 230–231, 233-234 Kubernetes Operator, 300–301 Professional Services team, 295 Quotas, 291–293 Recorder, 147, 149-152 scaling via automation, 301 use cases *APM (application performance* monitoring), 2 cloud and hybrid monitoring, 2 collaboration monitoring, 3 Internet and DNS monitoring, 3 network performance monitoring, 2

value proposition, 310–312 Views page, 123–124 throughput FTP. 235-238 maximum theoretical. 11–15 tools, Path Visualization, 133–134 topology view, 273 "Total Economic Impact of Cisco ThousandEyes End User Monitoring, The", 312 traditional troubleshooting, 125–127 transaction tests, 73-74, 147, 148 best practices, 154–155 exporting to ThousandEyes, 151–152 versus Page Load test, 148–149 script debugging, 153–154 markers, 154 setting up and creating with the Recorder, 149–152 Transactions view, 155–158 troubleshooting CLI utilities, 330 discards, 6–7 Enterprise Agents, 327–329 Linux-based agents, 329–330 load balancers, 241–243 logs, 323 methodology, 158 input, 159–161 output, 158–159 processes, 161–162 versus monitoring, 125 MSS (Maximum Segment Size), 244 - 248network path issues, 325-326 packet flooding, 248–253

packet loss, discards, 143–144 path visualization, reasons for loss result variation, 324–325 traditional, 125–127 transaction tests, 153–154 user experience, 239–240 **TTL (Time-to-Live), 129 tunnel** explicit, 136 implicit, 136 opaque, 136–137

# U

underlay network, 135, 138, 140–141
use cases
APM (application performance monitoring), 2
cloud and hybrid monitoring, 2
collaboration monitoring, 3
Internet and DNS monitoring, 3
network performance monitoring, 2
user account, creating, 275–276
user experience, 301
comparing, 304–305
troubleshooting, 239–240
utilization, Enterprise Agent, 39–41

# V

value proposition, 310–312 views device, 269–272 topology, 273 Transactions, 155–158 Views page, 123–124, 131–132 virtual appliance SSH, 330–332 VM (Virtual Machine) Agent, 44 voice layer, 76–77 RTP stream test, 80–82 SIP server test, 78–80 VXLAN, 307

### W

web applications, 148 web layer, 66 FTP server test, 74–77 HTTP server test, 66–70 Page Load test, 70–73 Transaction test, 73-74 WebDriver, 148 Webex, monitoring Cloud and Enterprise Agents, 218–219 Endpoint Agents, 218-221 webhooks, 105–107, 166–167 on GitHub, 296–297 options, 167-168 template, 167 testing, 167 widgets, 114-115 associated tests, 121 features, 120 iframe, 120 Wireshark, TCP SEQ tracking, 206-207

# X-Y-Z

### X.25, 5

Zoom, monitoring, 231 Cloud and Enterprise Agents, 232 Endpoint Agents, 232–234