



# Splunk Certified Study Guide

Prepare for the User, Power User, and  
Enterprise Admin Certifications

—  
Deep Mehta

Apress®

# **Splunk Certified Study Guide**

**Prepare for the User, Power User,  
and Enterprise Admin Certifications**

**Deep Mehta**

**Apress®**

## ***Splunk Certified Study Guide***

Deep Mehta  
Printserv, Mumbai, India

ISBN-13 (pbk): 978-1-4842-6668-7  
<https://doi.org/10.1007/978-1-4842-6669-4>

ISBN-13 (electronic): 978-1-4842-6669-4

Copyright © 2021 by Deep Mehta

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr  
Acquisitions Editor: Divya Modi  
Development Editor: Matthew Moodie  
Coordinating Editor: Divya Modi

Cover designed by eStudioCalamar

Cover image designed by Pixabay

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [booktranslations@springernature.com](mailto:booktranslations@springernature.com); for reprint, paperback, or audio rights, please e-mail [bookpermissions@springernature.com](mailto:bookpermissions@springernature.com).

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at [www.apress.com/978-1-4842-6668-7](http://www.apress.com/978-1-4842-6668-7). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*This book is dedicated to the late Mr. Amit Mahendra Mehta.  
This book would not have been possible without his blessing. Where I  
am today is because of him.*

# Table of Contents

<b>About the Author .....</b>	<b>xix</b>
<b>About the Technical Reviewer .....</b>	<b>xxi</b>
<b>Acknowledgments .....</b>	<b>xxiii</b>
<b>Introduction .....</b>	<b>xxv</b>
<b>Part I: Splunk Architecture, Splunk SPL (Search Processing Language), and Splunk Knowledge Objects .....</b>	<b>1</b>
<b>Chapter 1: An Overview of Splunk .....</b>	<b>3</b>
Overview of the Splunk Admin Exam .....	3
Structure.....	4
Requirements .....	4
Blueprint.....	6
An Introduction to Splunk .....	8
The History of Splunk .....	8
The Benefits of Splunk .....	9
The Splunk Architecture .....	9
Installing Splunk .....	13
Installing Splunk on macOS.....	13
Installing Splunk on Windows.....	16
Adding Data in Splunk.....	20
Summary.....	25
Multiple-Choice Questions .....	25
Further Reading .....	26

TABLE OF CONTENTS

- Chapter 2: Splunk Search Processing Language..... 27**
- The Pipe Operator ..... 28
- Time Modifiers ..... 28
- Understanding Basic SPL..... 29
  - Search Language Syntax..... 29
  - Boolean Operators in Splunk..... 30
  - Syntax Coloring in SPL ..... 31
- Sorting Results..... 31
  - Sort..... 31
- Filtering Commands..... 32
  - where ..... 32
  - dedup..... 33
  - head..... 33
  - tail..... 34
- Reporting Commands ..... 34
  - top ..... 34
  - rare..... 34
  - history ..... 35
  - table..... 35
  - stats..... 36
  - untable ..... 38
  - chart ..... 38
  - timechart ..... 39
- Filtering, Modifying, and Adding Fields ..... 40
  - eval..... 40
  - Rex..... 47
  - lookup..... 48
  - Field..... 49
- Grouping Results..... 49
  - Transaction ..... 50

Summary.....	50
Multiple-Choice Questions .....	51
References.....	52
<b>Chapter 3: Macros, Field Extraction, and Field Aliases .....</b>	<b>53</b>
Field Extraction in Splunk .....	54
Regular Expressions.....	54
Delimiters .....	57
Macros .....	58
Create a Macro Using Splunk Web .....	58
Create a Macro Using the .conf File .....	60
Field Aliases in Splunk.....	62
Setting up Field Aliases .....	62
Splunk Search Query .....	67
Summary.....	71
Multiple Choice Test Questions .....	72
References.....	73
<b>Chapter 4: Tags, Lookups, and Correlating Events .....</b>	<b>75</b>
Splunk Lookups .....	75
Looking up Table Files .....	77
Lookup Definitions.....	78
Automatic Lookups.....	79
Splunk Tags.....	81
Create Tags in Splunk Using Splunk Web .....	82
Tag Event Types in Splunk Web .....	83
Reporting in Splunk .....	85
Creating Reports in Splunk Web.....	85
Report Acceleration in Splunk .....	88
Scheduling a Report in Splunk .....	90

TABLE OF CONTENTS

- Alerts in Splunk..... 92
  - Create Alerts in Splunk Using Splunk Web ..... 92
  - Cron Expressions for Alerts ..... 94
- Summary..... 96
- Multiple-Choice Questions ..... 97
- References..... 99
  
- Chapter 5: Data Models, Pivot, and CIM ..... 101**
  - Understanding Data Models and Pivot..... 102
    - Datasets and Data Models..... 102
    - Creating Data Models and Pivot in Splunk ..... 102
  - Event Actions in Splunk ..... 112
    - GET Workflow Actions..... 112
    - Search Workflow Action..... 114
  - Common Information Model in Splunk..... 117
    - Defining CIM in Splunk ..... 117
  - Summary..... 121
  - Multiple-Choice Questions ..... 121
  - References..... 123
  
- Chapter 6: Knowledge Managers and Dashboards in Splunk..... 125**
  - Understanding the Knowledge Manager’s Role in Splunk ..... 125
    - Globally Transferring Knowledge Objects ..... 126
    - Enabling Knowledge Object Visibility..... 128
    - Restricting Read/Write Permissions on an App ..... 129
    - Orphaned Knowledge Objects ..... 130
  - Dashboards..... 132
    - Static Real-Time Dashboards ..... 133
    - Creating a Dashboard ..... 137
    - Adding a Report to a Dashboard..... 139
  - Dynamic Form-based Dashboards..... 140
    - Adding a Radio Button Using XML ..... 140
    - Adding a Time Modifier Using XML..... 142



Adding a Drop-Down Menu Using XML .....	145
Adding a Link List Using XML .....	147
Using the User Interface for Input .....	150
Summary.....	152
Multiple-Choice Questions .....	152
References.....	154
<b>Chapter 7: Splunk User/Power User Exam Set .....</b>	<b>155</b>
Questions .....	155
Summary.....	160
<b>Part II: Splunk Data Administration and System Administration .....</b>	<b>161</b>
<b>Chapter 8: Splunk Licenses, Indexes, and Role Management .....</b>	<b>163</b>
Buckets .....	163
How Does a Bucket Work? .....	164
How Search Is Performed in Buckets.....	165
Understanding journal.gz, .tsidx, and Bloom Filters.....	166
How Do Search Functions Work? .....	166
Splunk Licenses.....	167
Changing a License Group in Splunk.....	168
Managing Splunk Licenses .....	169
License Masters and Slaves .....	169
Adding a License in Splunk .....	171
License Pooling.....	172
Creating a License Pool .....	172
Managing Indexes in Splunk.....	172
Creating an Index in Splunk.....	173
User Management.....	177
Adding a Native User .....	177
Defining Role Inheritance and Role Capabilities.....	179

TABLE OF CONTENTS

- Summary..... 181
- Multiple-Choice Questions ..... 182
- References..... 184
- Chapter 9: Machine Data Using Splunk Forwarder and Clustering ..... 185**
  - Splunk Universal Forwarder..... 186
    - Configuring Splunk Indexer to Listen to Data for Universal Forwarder ..... 186
    - Configuring Windows Splunk Forwarder ..... 187
    - Splunk Universal Forwarder Using Windows..... 187
    - Splunk Universal Forwarder Using .msi ..... 188
    - Configuring Linux Splunk Forwarder ..... 189
  - Splunk’s Light and Heavy Forwarders ..... 190
    - Splunk Heavyweight Forwarder ..... 191
    - Splunk Light Forwarder ..... 192
  - Forwarder Management..... 193
    - Configuring Forwarder Management..... 193
    - Configuring the Forwarder Management Client ..... 195
  - Splunk Indexer Clusters ..... 195
    - Configuring Indexer Clusters ..... 196
  - Splunk Lightweight Directory Access Protocol (LDAP)..... 201
    - Creating an LDAP Strategy ..... 202
    - Mapping LDAP Group to Splunk Roles..... 205
  - Splunk Security Assertion Markup Language (SAML)..... 206
    - Configuring Splunk SAML..... 206
    - Map SAML to User Roles ..... 209
- Summary..... 210
- Multiple-Choice Questions ..... 210
- References..... 212

<b>Chapter 10: Advanced Data Input in Splunk</b> .....	<b>213</b>
Compress the Data Feed.....	214
Indexer Acknowledgment.....	215
Securing the Feed.....	215
Queue Size .....	216
Monitor Input.....	217
Monitor Files.....	217
Monitor Directories.....	218
Monitor Files and Directory Using Splunk Web .....	218
Monitor File and Directory Using inputs.conf .....	220
Scripted Input .....	221
Scripted Input Using Splunk Web .....	222
Scripted Input Using inputs.conf file .....	224
Network Input .....	226
Add Network Input Using Splunk Web and Deploy It to the Forwarder .....	226
Modify Network Input Using .conf Files.....	228
Pulling Data Using Agentless Input .....	231
HTTP Input Using Splunk Web .....	231
Configure HTTP Event Collector in Splunk.....	234
Configure HTTP Input Using .conf File .....	235
Configure HTTP Event Collector in Splunk Using .conf File .....	236
Parse Data in Splunk Using HTTP Event Collector .....	237
Summary.....	238
Multiple-Choice Questions .....	238
References .....	241
<b>Chapter 11: Splunk's Advanced .conf File and Diag</b> .....	<b>243</b>
Understanding Splunk .conf files .....	243
props.conf .....	244
indexes.conf .....	246
transforms.conf .....	246
inputs.conf.....	247

TABLE OF CONTENTS

- outputs.conf ..... 248
- deploymentclient.conf ..... 250
- Setting Fine-Tuning Input..... 250
  - Custom Source Types Using Splunk Web ..... 251
  - Custom Source Types Using props.conf ..... 252
- Anonymizing the Data..... 253
  - props.conf to Anonymize Data with a sed Script..... 254
  - props.conf and transforms.conf to Anonymize Data with Regular Expressions..... 255
- Understanding Merging Logic in Splunk ..... 256
  - Configuration File Precedence..... 257
  - Splunk .conf Files Location ..... 257
  - Configuration Merging Logic ..... 258
- Debugging Configuration Files ..... 260
  - Example: Btool for Troubleshooting a Configuration File ..... 260
- Creating a Diag ..... 261
  - Creating a Diag in Splunk..... 261
- Summary..... 263
- Multiple-Choice Questions ..... 263
- Reference..... 265
- Chapter 12: Splunk Admin Exam Set..... 267**
  - Questions ..... 267
  - Summary..... 272
- Part III: Advanced Splunk ..... 273**
- Chapter 13: Infrastructure Planning with Indexer and Search Head Clustering .... 275**
  - Capacity Planning for Splunk Enterprise..... 276
    - Dimensions of a Splunk Enterprise Deployment ..... 276
    - Disk Storage for Splunk Enterprise ..... 277

Configuring a Search Peer .....	278
Configuring a Search Peer from Splunk Web .....	278
Configure Splunk Search Peer from the .conf File .....	279
Configure Search Peer from Splunk CLI .....	279
Configure a Search Head .....	280
Configuring a Search Head Using Splunk Web .....	282
Configure Splunk Search Head Using .conf file .....	282
Configuring a Search Head from Splunk CLI .....	283
Search Head Clustering .....	283
Search Head Cluster Captain .....	285
The Role of Captains .....	285
Captain Election .....	285
Configure Search Head Cluster Using CLI in Splunk .....	286
Multisite Indexer Clustering .....	287
Configure Multisite Indexer Clustering Using .conf Files .....	288
Configure Splunk Multisite Indexer Clustering Using CLI .....	290
Splunk Validated Architectures (SVAs) .....	294
Designing Splunk Validated Architectures .....	294
Splunk Architecture Practices .....	303
Use Case: Company XYZ .....	303
Splunk Data Inputs .....	305
Splunk Index Calculation .....	307
Splunk Total Disk Size .....	308
Splunk User Planner .....	309
Hardware and Splunk Scaling Considerations .....	310
Disk Size Calculation .....	311
Summary .....	313
Multiple-Choice Questions .....	313
References .....	315

TABLE OF CONTENTS

- Chapter 14: Troubleshooting in Splunk ..... 317**
- Monitoring Console ..... 317
  - Single Instance Deployment Monitoring Console ..... 318
  - Multi-Instance Deployment Monitoring Console ..... 318
  - Monitor System Health with Monitoring Console ..... 321
  - Configure Forwarder Monitoring for the Monitoring Console ..... 323
- Log Files for Troubleshooting ..... 324
- The metrics.log File ..... 325
  - Pipeline Messages..... 327
  - Queue Messages ..... 328
  - Thruput Messages..... 328
  - Tcpout Connection Messages..... 329
  - udpin\_connections Messages..... 329
  - bucket\_metrics Messages ..... 330
- Job Inspector ..... 331
  - Job Inspector Example Query..... 331
- Troubleshooting License Violations ..... 334
  - Violation Due to an Improper Connection Between License Master and Slave Node..... 334
- Troubleshooting Deployment Issues ..... 335
  - Troubleshooting Splunk Forwarders..... 336
  - Troubleshooting Splunk Indexers ..... 336
- Troubleshooting Clustering Issues ..... 337
  - Multi-Search Affinity..... 338
  - More Bucket Issues ..... 338
- Summary..... 338
- Multiple-Choice Questions ..... 339
- References ..... 341

<b>Chapter 15: Advanced Deployment in Splunk .....</b>	<b>343</b>
Deploying Apps Through the Deployment Server.....	343
Create App Directories and View Apps in Forwarder Management.....	344
Redeploy Apps to the Client.....	346
App Management Issues .....	347
Creating a Server Group Using ServerClass.conf.....	347
Configure a Server Class Using Forwarder Management.....	348
Deploy Configuration File Through Cluster Master.....	349
Managing Indexes on Indexer Using Master Node .....	349
Deploy App on Search Head Clustering.....	352
Configure the Deployer to Distribute Search Head Apps .....	352
Load Balancing .....	355
Configure Static Load Balancing in Splunk Forwarder Using outputs.conf.....	356
Configure a Static Load Balancer by Time.....	356
Specify Load Balancing from Splunk CLI.....	357
Indexer Discovery .....	357
Configure Indexer Delivery .....	357
SOCKS Proxy .....	360
Configure SOCKS Proxy .....	360
Summary.....	361
Multiple-Choice Questions .....	362
References.....	363
<b>Chapter 16: Advanced Splunk .....</b>	<b>365</b>
Managing Indexes.....	365
Configure Event Indexes.....	366
Configure Metrics Indexes.....	367
Remove Indexes and Index Data for Managing Indexes.....	368
Configure Index Parallelization for Managing Indexes .....	368
Manage Index Storage .....	369
Move the Index Database .....	369

TABLE OF CONTENTS

- Configure Maximum Index Size for Indexer Storage ..... 370
- Set Limit for Disk Usage in Splunk ..... 371
- Managing Index Cluster ..... 372
  - Configuring Peer Node to Offline ..... 372
  - Configure Splunk to Maintenance Mode Using Splunk CLI ..... 373
  - Rolling Restart in Splunk Using Splunk CLI ..... 374
  - Remove Excess Buckets Copies from the Indexer Cluster ..... 375
  - Remove a Peer from Master’s List ..... 375
- Managing a Multisite Index Cluster..... 376
  - Master Site in Multisite Index Cluster Fails ..... 376
  - Restart Indexing in the Multisite Cluster After a Master Restart or Site Failure..... 377
  - Move a Peer to a New Site ..... 377
- REST API Endpoints..... 378
  - Running Searches Using REST API ..... 379
- Splunk SDK ..... 381
  - Python Software Development Kit for Splunk ..... 381
- Summary..... 385
- Multiple-Choice Questions ..... 385
- References..... 387
- Chapter 17: Final Practice Set ..... 389**
  - Questions ..... 389
  - Summary..... 394
- Chapter 18: Setting up a Splunk Environment with AWS ..... 395**
  - Amazon Web Services..... 395
    - Configuring an EC2 Instance Using the AWS Management Console..... 396
    - Configuring Splunk on an EC2 Instance ..... 399
  - Deploying Multisite Index Clustering..... 401
    - Configuring a Cluster Master..... 401
    - Configuring a Slave Node ..... 402



Deploying a Search Head .....	403
Configuring a Search Head .....	403
Configuring Search Head Clustering .....	404
Deploying Configurations .....	406
Configuring a Cluster Master .....	407
Deploying an App to Search Head Cluster Using Deployment Server .....	409
Configuring a Universal Forwarder for Indexer Discovery .....	411
Configuring a Universal Forwarder for Indexer Deployment .....	413
Configuring the Master Node, Deployment Server, and Search Head 4 to Send Internal Logs .....	416
Monitoring Distributed Environments .....	417
Adding a Search Peer to Monitor .....	417
General Setup for Distributed Environments .....	418
Conclusion .....	419
<b>Index .....</b>	<b>421</b>

# About the Author



**Deep Mehta** is an AWS Certified Associate Architect (ongoing), Docker Certified Associate, Certified Splunk Architect, and Certified Splunk User, Power User, and Admin. He has worked with the Splunk platform since 2017 and has related consulting experience in the telecommunication, aviation, and health care industries. In addition to his passion for big data technologies, he loves playing squash and badminton.

# About the Technical Reviewer



**James Miller** is an innovator and senior project lead and solution architect with 37 years of extensive design and development experience in multiple platforms and technologies. He leverages his consulting experience to provide hands-on leadership in all phases of advanced analytics and related technology projects. His work includes recommendations on process improvement, report accuracy, adoption of disruptive technologies, enablement, insight identification, statistics for data science, predictive analytics, big data visualization, Watson analytics, and implementing and mastering Splunk.

# Acknowledgments

I am grateful to all my readers for choosing this book. I am also grateful to my parents, Paresh and Rupa Mehta, and my family and friends for their support.

I would like to thank Sanket Sheth of Elixia Tech. for providing the dataset, and a very special thanks to Neha Doshi for extending her support and providing guidance.

# Introduction

Splunk is a software technology for monitoring, searching, analyzing, and visualizing machine-generated data in real time. This book discusses the roles of a Splunk admin and explains how Splunk architecture can be efficiently deployed and managed. It covers everything you need to know to ace the Splunk exams. The book is written to be used interactively and includes practice datasets and test questions at the end of every chapter.

This book is divided into four modules, and three modules are dedicated to the exam.

The first module comprises six chapters dedicated to passing the Splunk Core Certified User exam and the Splunk Core Certified Power User exam. It covers installing Splunk, Splunk's Search Processing Language (SPL), field extraction, field aliases and macros in Splunk, creating Splunk tags, Splunk lookups, and invoking alerts. You learn how to make a data model and prepare an advanced dashboard in Splunk.

The second module is dedicated to the Splunk Enterprise Certified Admin exam and consists of four chapters. It covers Splunk licenses and user role management, Splunk forwarder configuration, indexer clustering, Splunk security policies, and advanced data input options.

The third module also focuses on the Splunk Enterprise Certified Admin exam, but its chapters teach admins to troubleshoot and manage the Splunk infrastructure.

In the fourth module, you learn how to set up Splunk Enterprise on the AWS platform, and you are introduced to some of the best practices in Splunk.

At the end of every chapter is a multiple-choice test to help candidates become more familiar with the exam.

# **PART I**

# **Splunk Architecture, Splunk SPL (Search Processing Language), and Splunk Knowledge Objects**

## CHAPTER 1

# An Overview of Splunk

Splunk is a software technology for monitoring, searching, analyzing, and visualizing machine-generated data in real-time. This tool can monitor and read several types of log files and store data as events in indexers. It uses dashboards to visualize data in various forms.

This chapter discusses the basics of Splunk, including its history and architecture, and delves into how to install the software on local machines. You see the layout of the Splunk Enterprise Certified Admin exam. And, you learn how to add user data and a props.conf file, and you learn the process of editing timestamps, which is useful in the later chapters. A few sample questions are at the end of the chapter.

Summing it up, this chapter covers the following topics.

- An overview of the Splunk Enterprise Certified Admin exam
- An introduction to Splunk
- The Splunk architecture
- Installing Splunk on macOS and Windows
- Adding data to Splunk

## Overview of the Splunk Admin Exam

A Splunk Enterprise Certified Admin is responsible for the daily management of Splunk Enterprise, including performance monitoring, security enhancement, license management, indexers and search heads, configuration, and adding data to Splunk. The following are the areas of expertise that the exam tests.

- Splunk deployment
- License management
- Splunk applications

- Splunk configuration files
- Users, roles, and authentication
- Adding data
- Distributed searches
- Splunk clusters
- Deploying forwarders with forwarder management
- Configuring common Splunk data inputs
- Customizing the input parsing process

In the next section, you learn about the admin exam's structure.

## Structure

The Splunk Enterprise Certified Admin exam is in multiple-choice question format. You have 57 minutes to answer 63 questions and an additional 3 minutes to review the exam agreement, totaling 60 minutes. The passing score for this exam is 75%. The exam's registration fee is \$120 (USD). Refer to [www.splunk.com/pdfs/training/Splunk-Test-Blueprint-Admin-v.1.1.pdf](http://www.splunk.com/pdfs/training/Splunk-Test-Blueprint-Admin-v.1.1.pdf) for more information.

The exam questions come in three formats.

- **Multiple choice:** You must select the option that is the best answer to a question or to complete a statement.
- **Multiple responses:** You must select the options that best answer a question or completes a statement.
- **Sample directions:** You read a statement or question and select only the answer(s) that represent the most ideal or correct response.

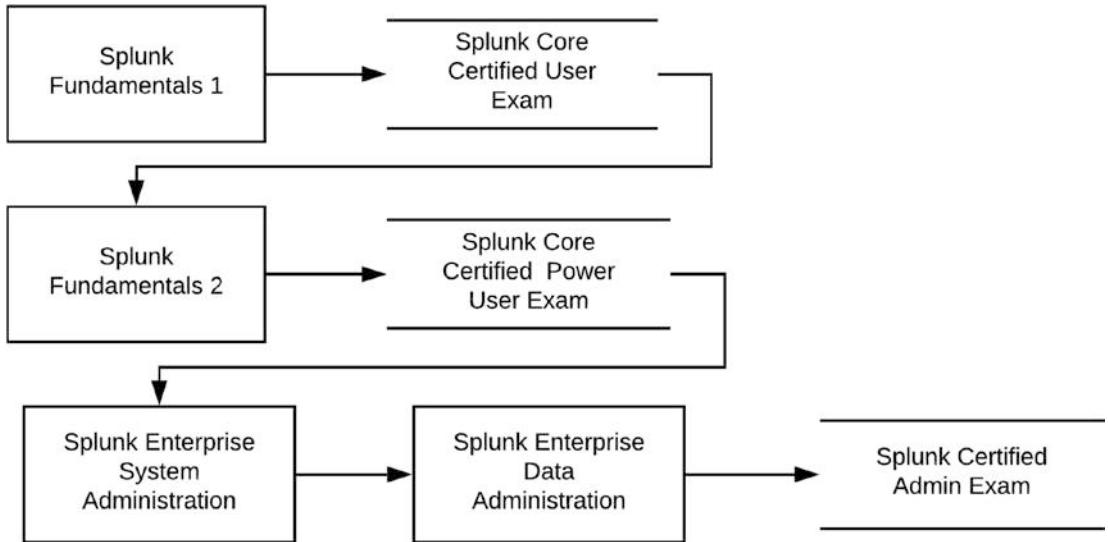
## Requirements

The Splunk Enterprise Certified Admin exam has two prerequisites. You must first pass the following exams.



- The Splunk Core Certified Power User exam
- The Enterprise System Administration and Splunk Data Administration courses

Four courses support these exams. The learning flow is shown in Figure 1-1.



**Figure 1-1.** *Splunk exam prerequisites*

- **Splunk Fundamentals 1** is offered to students in two ways: e-learning or instructor-led. This course introduces you to the Splunk platform.
- **Splunk Core Certified User Exam** tests your knowledge of and skills in searching, using fields, creating alerts, using lookups, and creating basic statistical reports and dashboards.
- **Splunk Fundamentals 2** is an instructor-led course on searching and reporting commands and creating knowledge objects.
- The **Splunk Core Certified Power User exam** tests the knowledge and skills required for SPL searching and reporting commands and building knowledge objects, using field aliases and calculated fields, creating tags and event types, using macros, creating workflow actions and data models, and normalizing data with the Common Information Model.

- **Splunk Enterprise System Administration** is instructor-led and designed for system administrators responsible for managing the Splunk Enterprise environment. The course teaches fundamental information for Splunk license managers, indexers, and search heads.
- **Splunk Enterprise Data Administration** is instructor-led and designed for system administrators responsible for adding remote data to Splunk indexers. The course provides fundamental information on Splunk forwarders and methods.
- The **Splunk Enterprise Certified Admin** exam tests your knowledge of and skills in managing various components of Splunk Enterprise, including license management, indexers and search heads, configuration, monitoring, and adding data to Splunk.

Modules 2 and 3 of this book focus on the Splunk Enterprise system administration and data administration exams.

## Blueprint

The Splunk Enterprise Certified Admin exam has 17 sections, described as follows.

- **Section 1: Splunk Admin Basics (5%)** This section focuses on identifying Splunk components.
- **Section 2: License Management (5%)** This section focuses on identifying license types and understanding license violations.
- **Section 3: Splunk Configuration Files (5%)** This section focuses on configuration layering, configuration precedence, and the Btool command-line tool to examine configuration settings.
- **Section 4: Splunk Indexes (10%)** This section focuses on basic index structure, types of index buckets, checking index data integrity, the workings of the indexes.conf file, fish buckets, and the data retention policy.
- **Section 5: Splunk User Management (5%)** This section focuses on user roles, creating a custom role, and adding Splunk users.

- **Section 6: Splunk Authentication Management (5%)** This section focuses on LDAP, user authentication options, and multifactor authentication.
- **Section 7: Getting Data In (5%)** This section focuses on basic input settings, Splunk forwarder types, configuring the forwarder, and adding UF input using CLI.
- **Section 8: Distributed Search (10%)** This section focuses on distributed search, the roles of the search head and search peers, configuring a distributed search group, and search head scaling options.
- **Section 9: Getting Data In—Staging (5%)** This section focuses on the three phases of the Splunk indexing process and Splunk input options.
- **Section 10: Configuring Forwarders (5%)** This section focuses on configuring forwarders and identifying additional forwarder options.
- **Section 11: Forwarder Management (10%)** This section focuses on deployment management, the deployment server, managing forwarders using deployment apps, configuring deployment clients, configuring client groups, and monitoring forwarder management activities.
- **Section 12: Monitor Inputs (5%)** This section examines your knowledge of file and directory monitor inputs, optional settings for monitor inputs, and deploying a remote monitor input.
- **Section 13: Network and Scripted Inputs (5%)** This section examines your knowledge of the network (TCP and UDP) inputs, optional settings for network inputs, and a basic scripted input.
- **Section 14: Agentless Inputs (5%)** This section examines your knowledge of Windows input types and the HTTP event collector.
- **Section 15: Fine-Tuning Inputs (5%)** This section examines your knowledge of the default processing during the input phase and configuring input phase options, such as source type fine-tuning and character set encoding.

- **Section 16: Parsing Phase and Data (5%)** This section examines your knowledge of the default processing during parsing, optimizing, and configuring event line breaking, extraction of timestamps and time zones from events, and data preview to validate event created during the parsing phase.
- **Section 17: Manipulating Raw Data (5%)** This section examines your knowledge of how data transformations are defined and invoked and the use of transformations with `props.conf` and `transforms.conf` and `SEDCMD` to modify raw data.

## An Introduction to Splunk

The word *splunk* comes from the word *spelunking*, which means to explore caves. Splunk can analyze almost all known data types, including machine data, structured data, and unstructured data. Splunk provides operational feedback on what is happening across an infrastructure in real time—facilitating fast decision-making.

Splunk is commonly thought of as “a Google for log files” because, like Google, you can use Splunk to determine the state of a network and the activities taking place within it. It is a **centralized log management tool**, but it also works well with structured and unstructured data. Splunk **monitors, reports, and analyzes real-time machine data**, and indexes data based on timestamps.

## The History of Splunk

Splunk was founded by Rob Das, Erik Swan, and Michael Baum in October 2003. It grew from a small startup company to one of the biggest multinational corporations for security information and event management (SIEM) tools. Before Splunk, a business needing to troubleshoot its environment had to rely on the IT department, where a programmer wrote scripts to meet needs. This script ran on top of a platform to generate a report.

As a result, companies didn’t have a precise way to discover problems deep inside their infrastructure. Splunk was created to deal with this issue. Initially, Splunk focused on analyzing and understanding a problem, learning what organizations do when something goes wrong, and retracing footprints.

The first version of Splunk was released in 2004 in the Unix market, where it started to gain attention.

It is important to understand why this software was developed. The following section discusses Splunk's many useful benefits.

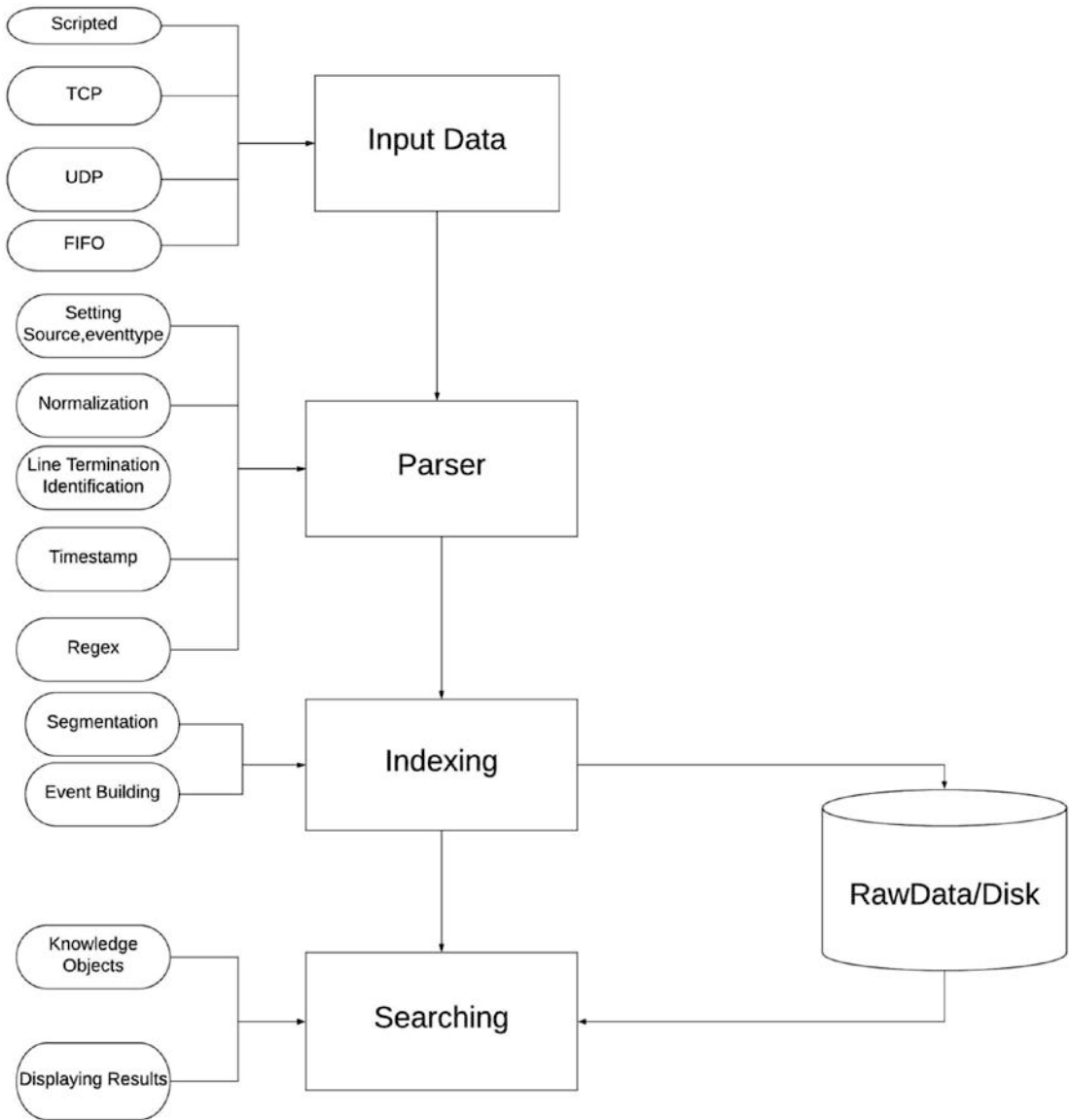
## The Benefits of Splunk

Splunk offers a variety of benefits, including the following.

- Converts complex log analysis report into graphs
- Supports structured as well as unstructured data
- Provides a simple and scalable platform
- Offers a simple architecture for the most complex architecture
- Understands machine data
- Provides data insights for operational intelligence
- Monitors IT data continuously

## The Splunk Architecture

The Splunk indexer works in a specified manner in a set architecture (see Figure 1-2).



**Figure 1-2.** Splunk architecture diagram

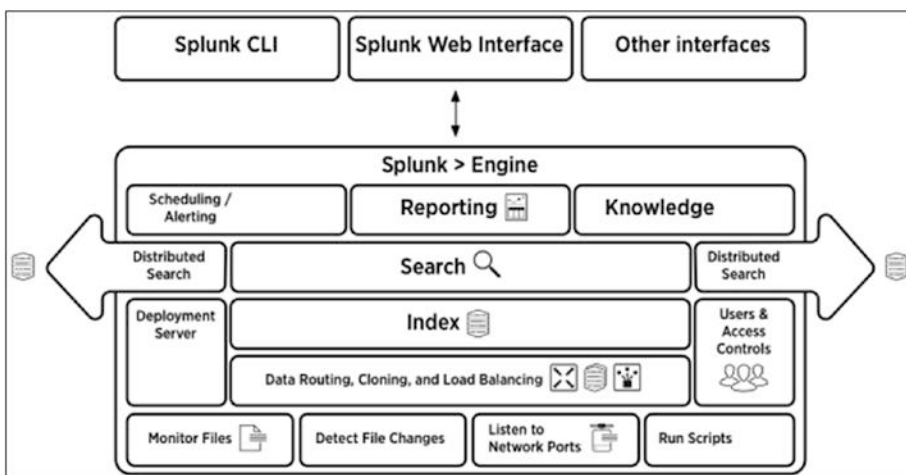
Let’s parse this diagram and introduce its components.

- **Input data:** This is the first phase of onboarding data. There are several methods to bring data into Splunk: it can listen to your port, your REST API endpoint, the Transmission Control Protocol (TCP), the User Datagram Protocol (UDP), and so on, or use scripted input.

- **Parser:** The second phase is to parse the input, in which a chunk of data is broken into various events. The maximum size of the data in the parsing pipeline is 128 MB. In the parsing phase, you can extract default fields, such as the source type. You can also extract timestamps from the data, identify the line's termination, and perform other similar actions. You can also mask sensitive but useful data. For example, if the data is from a bank and includes a customers' account numbers, masking data is essential. In the parsing phase, you can apply custom metadata, if required.
- **Indexing:** In this phase, the event is broken into segments within which searching can be done. The data is written to disk, and you can design indexing data structures.
- **Searching:** In this phase, the search operations are performed on top of the index data, and you can create a knowledge object and perform any task; for example, a monthly sales report.

The input data, the parser, and the indexer are all on one standalone machine. In contrast to this, in a distributed environment, the input data is parsed to the indexer or the heavy forwarder using Universal Forwarder (UF), which is a lightweight program that gets data in Splunk. In the UF, you cannot search for data or perform any operation. You look at this later in the chapter.

Figure 1-3 shows Splunk's architecture.



**Figure 1-3.** *Splunk's architecture*

The following tasks can be performed in the Splunk architecture.

- You can **receive data** through network ports and detect file changes in real time.
- You can monitor files and detect files in real time.
- You can run scripts to get **customized data**.
- Data routing, cloning, and load balancing are available in **distributed environments**, which you learn about in later chapters.
- **User access controls** preserve security. There are various security levels: user, power user, and admin. Users can write or read indexes based on their rights.
- The deployment server manages an entire deployment for the stack. You can deploy new applications using the **deployment server**.
- When an indexer receives data from a parser, it indexes it, and you can break down the event into **segments**.
- Once the data is stored in the indexer, you can perform **search operations**.
- You can do a scheduled search on indexed data.
- You can **generate a data alert** by setting parameters; for example, when the transaction time exceeds 15 minutes in a particular transaction.
- You can **create reports** by scheduling, saving searches, or creating a macro. There are a variety of ways to generate reports.
- Knowledge objects are useful for creating specialized reports from user-defined data, unstructured data, and so on.
- You can **access the Splunk instance** using either the Splunk web interface, which is the most popular option, or the Splunk CLI.

Now let's move forward to learn how to get Splunk quickly installed.



# Installing Splunk

You can download and install Splunk Enterprise for free using its 60-day trial version that indexes 500 MB/day. All you need to do is create an account at [www.splunk.com/en\\_us/download/splunk-enterprise.html](http://www.splunk.com/en_us/download/splunk-enterprise.html).

After 60 days, you can convert to a perpetually free license or purchase a Splunk Enterprise license to continue using the expanded functionality designed for enterprise-scale deployments. (Chapter 8 discusses Splunk licenses in detail).

Table 1-1 shows the relationship of a few Splunk attributes to their default ports during installation. The importance of each attribute is discussed later in the book.

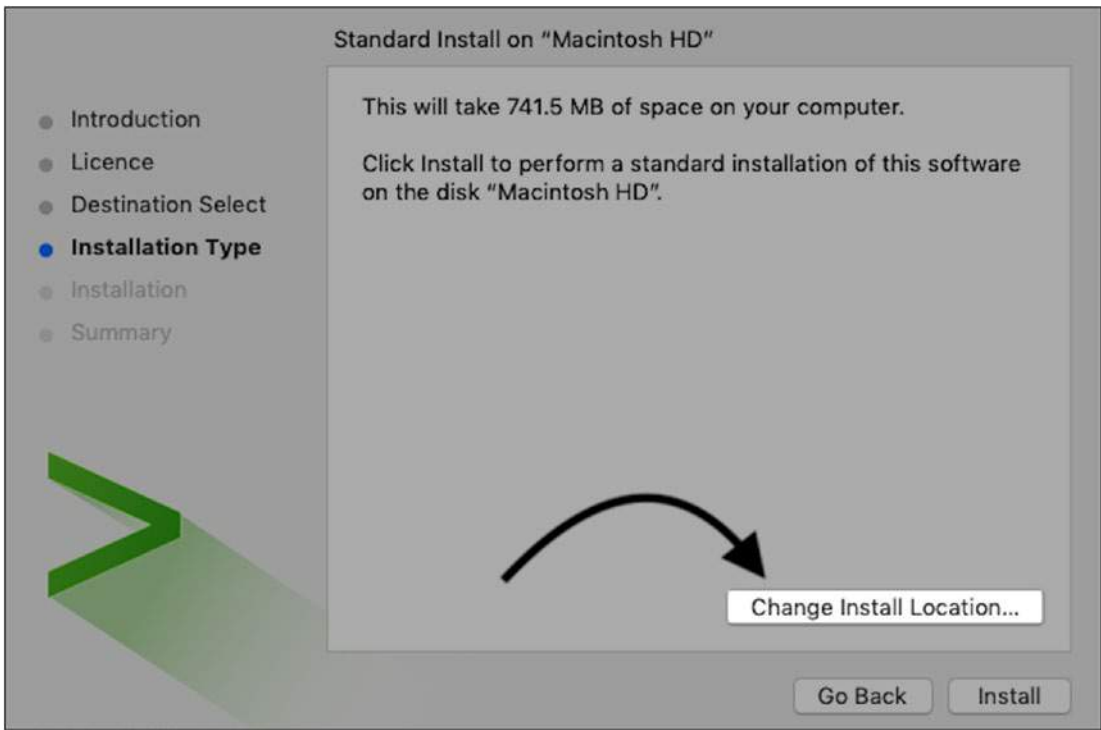
*Table 1-1. Splunk Attributes & Default Port Values*

Attribute	Default Port
Splunk default port	8000
Splunk management port	8089
Splunk KV Store	8191

## Installing Splunk on macOS

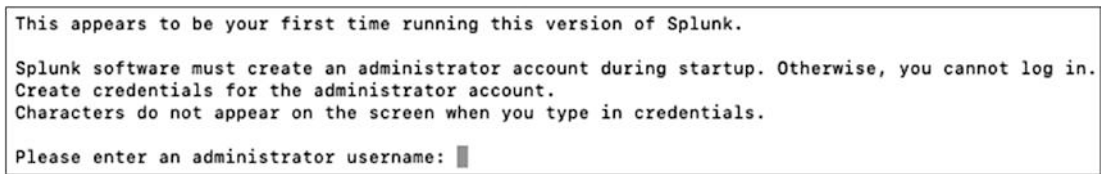
macOS users should follow these steps to install Splunk.

1. Sign in to your Splunk account.
2. Download the Splunk file at [www.splunk.com/en\\_us/download/splunk-enterprise.html#tabs/macos](http://www.splunk.com/en_us/download/splunk-enterprise.html#tabs/macos).
3. Open the downloaded file.
4. Click the Install button.
5. Click the Continue button in the next step.
6. Click the Continue button until you are prompted to click the Install button.
7. If you want to change the install location first, you can do it by clicking Change Install Location (see Figure 1-4).



**Figure 1-4.** *Installing Splunk*

8. After selecting the path in which you want to install Splunk, click the Install button.
9. Enter **admin** as the administrator username (see Figure 1-5).



**Figure 1-5.** *Create administrator User*

10. Enter your new password, as shown in Figure 1-6.

```

This appears to be your first time running this version of Splunk.

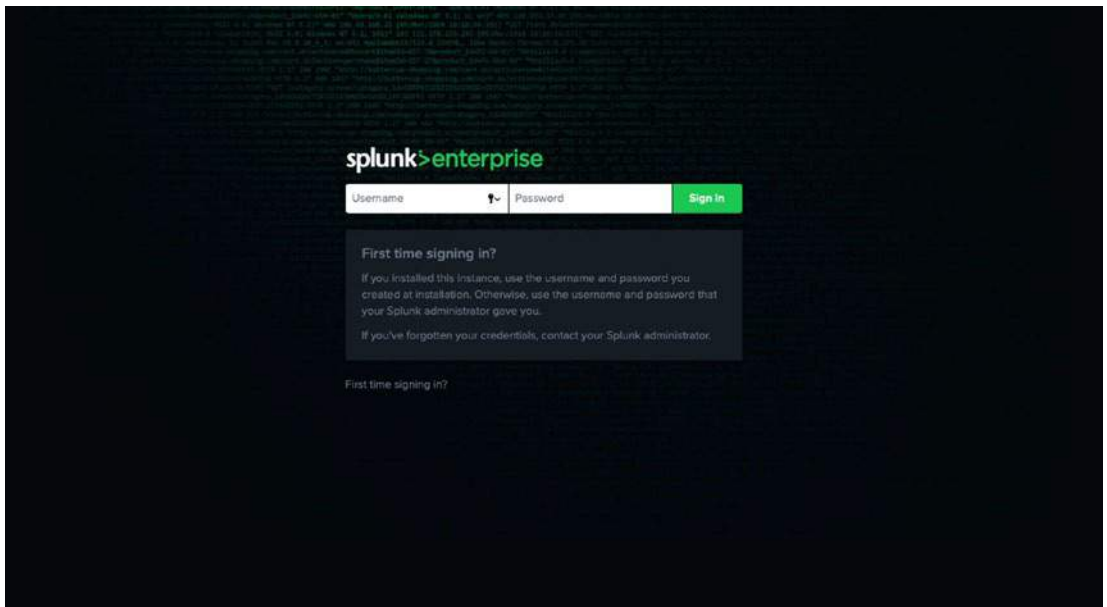
Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: admin
Password must contain at least:
  * 8 total printable ASCII character(s).
Please enter a new password: █

```

**Figure 1-6.** Create password for administrator User

11. Start Splunk at **http://localhost:8000**. Enter the username and password that you just created (Figure 1-7).



**Figure 1-7.** Splunk User Interface

12. Once you are logged in, go to the Search & Reporting app and enter the following Splunk processing command to test it.

```
index = "_audit"
```

If you get a response, you have set up the installation successfully. You should see a screen similar to Figure 1-8.

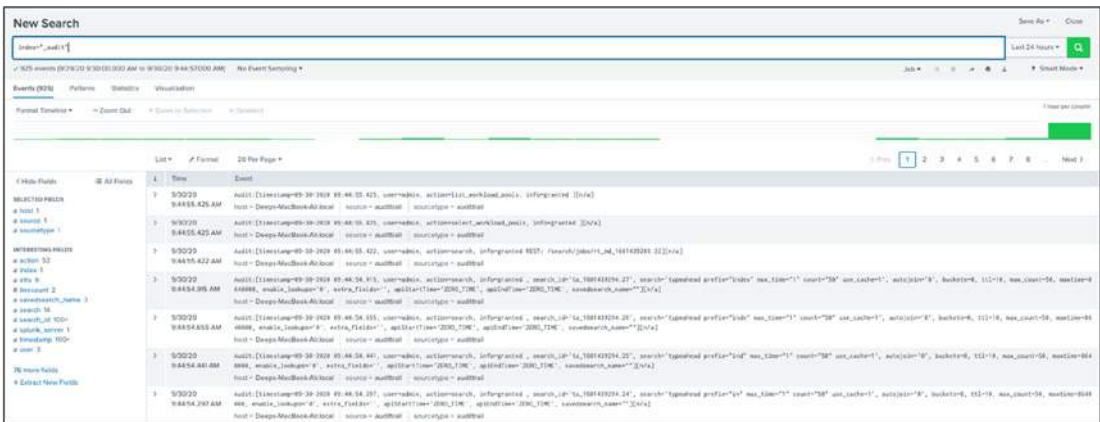


Figure 1-8. Splunk Events for index audit

Note that host, index, linecount, punct, source, sourcetype, splunk\_server, and timestamp are a few default fields added to Splunk when indexing your data source.

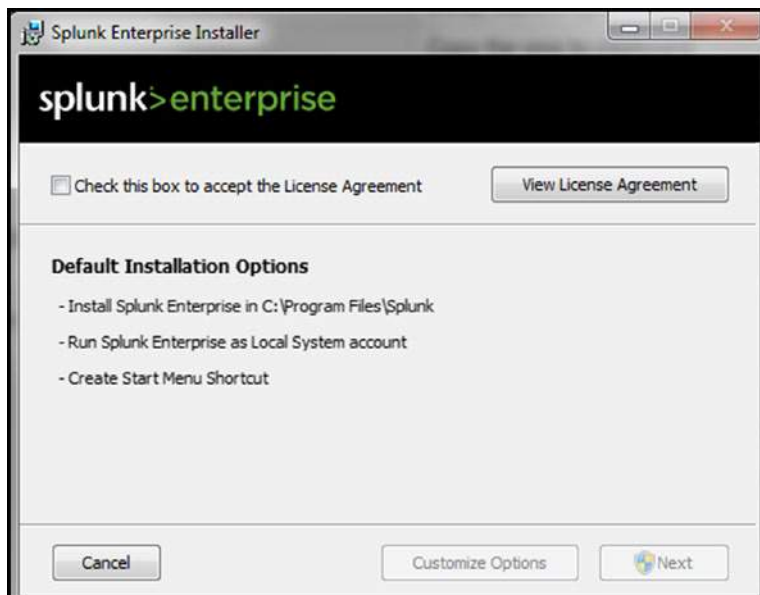
This sums up the entire process of installing Splunk onto macOS.

Next, let’s discuss how to install it on the Windows operating system.

## Installing Splunk on Windows

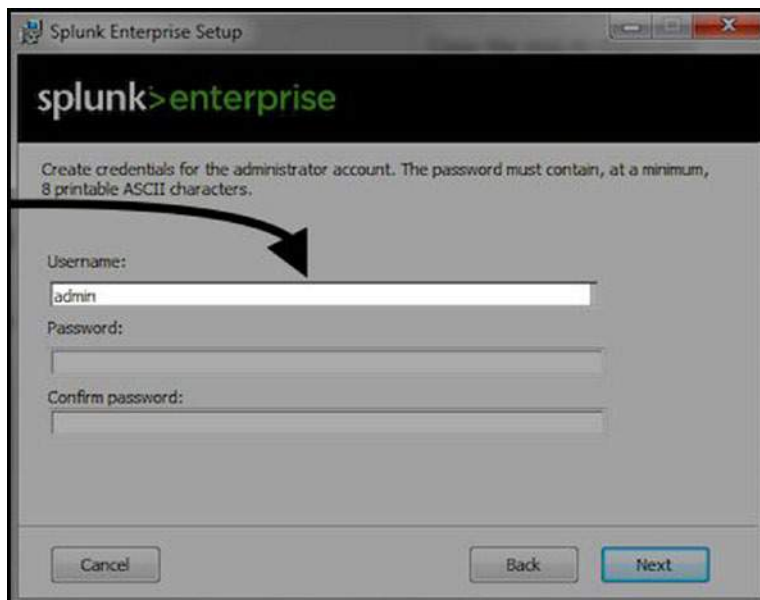
Windows users should follow these steps to install Splunk.

1. Sign in to your Splunk account.
2. Download the Splunk file from [www.splunk.com/en\\_us/download/splunk-enterprise.html](http://www.splunk.com/en_us/download/splunk-enterprise.html).
3. Open the downloaded file. Your screen should look similar to Figure 1-9.



**Figure 1-9.** *Installing Splunk*

4. Click the check box to accept the license agreement.
5. Enter **admin** as the username, as shown in Figure 1-10.



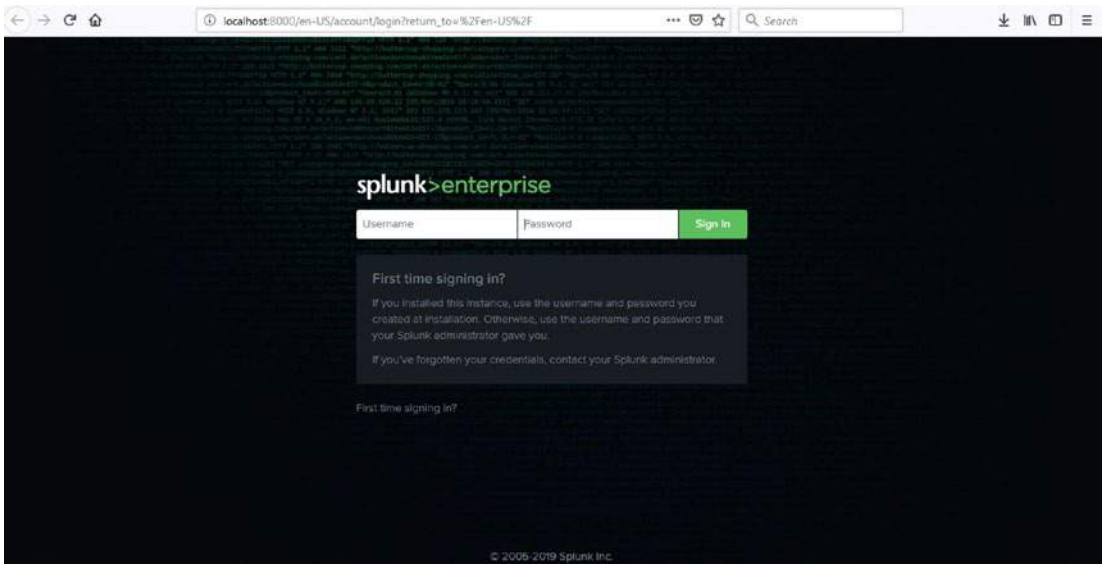
**Figure 1-10.** *Create administrator User*

6. Enter your password and confirm it. Then, click Next to proceed to the next step (see Figure 1-11).



**Figure 1-11.** Create Password for administrator User

7. Click the Install button to install Splunk on your local machine.
8. Go to **http://localhost:8000**. In the Splunk login screen (see Figure 1-12), enter the username and password that you used in steps 5 and 6.

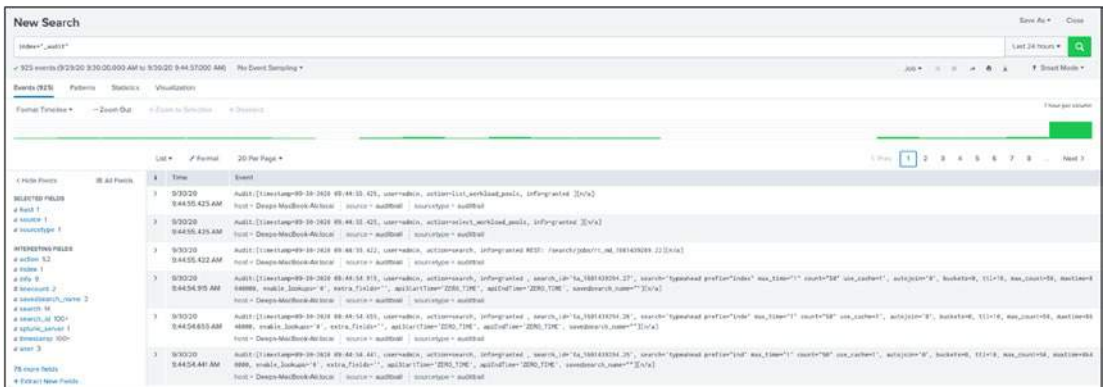


**Figure 1-12.** *Splunk User Interface*

- Once you are logged in, go to the Search & Reporting app and enter the following Splunk processing command to test it.

"index as "\_audit"

If you get a response, you have set up the installation successfully. You should see a screen similar to Figure 1-13.



**Figure 1-13.** *Splunk Events for index audit*

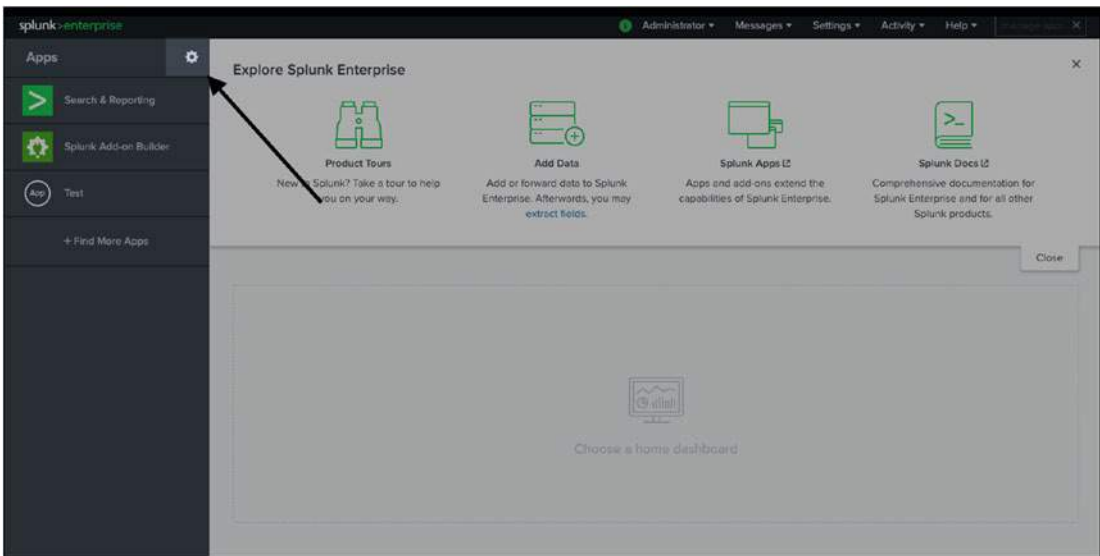
Note that `host`, `index`, `linecount`, `punct`, `source`, `sourcetype`, `splunk_server`, and `timestamp` are a few default fields added to Splunk when indexing your data source.

With this, you have learned how to install Splunk on both macOS and Windows systems. You also learned about Splunk and its architecture and the Splunk Enterprise Certified Admin exam. In the last section of this chapter, you learn the process of adding data in Splunk.

## Adding Data in Splunk

Once Splunk is installed on your local machine, the next task is to onboard data. To do this, you need to create a new app named `test` to carry out your tasks.

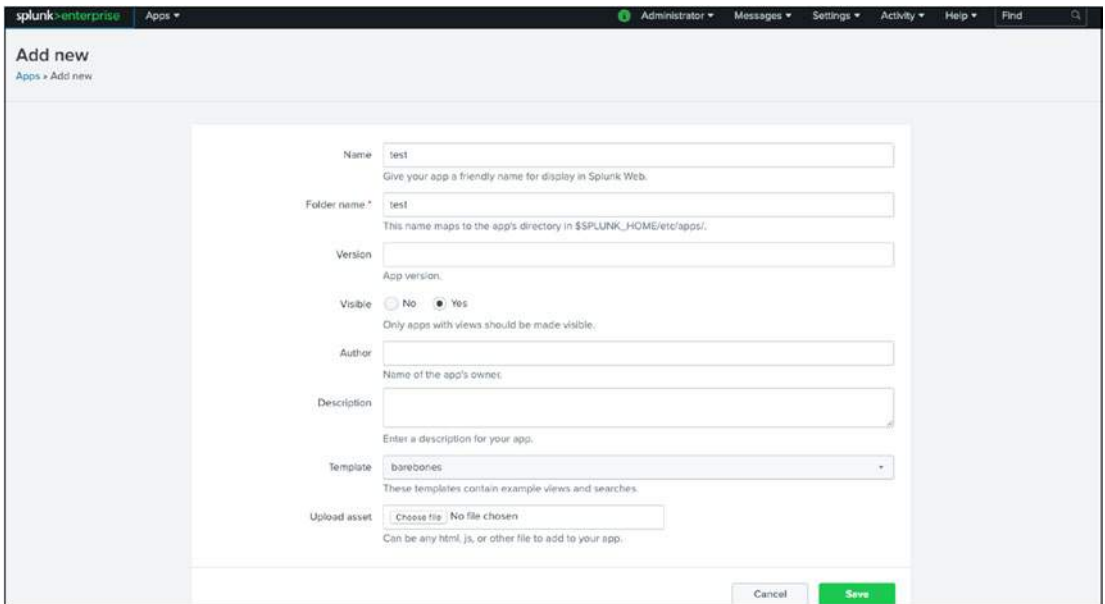
1. To create a new app in Splunk, click the gear icon next to Apps, as shown in Figure 1-14.



**Figure 1-14.** Splunk test App

2. Click Create. In this case, the app's name is `test` and the folder's name is also `test` (see Figure 1-15). This folder resides in `$SPLUNK_HOME/etc/apps/test`.





**Figure 1-15.** Create app Test: Splunk Web

3. Once the Splunk application is created, open any text editor and create a `props.conf` file in `$SPLUNK_HOME/etc/apps/test/local`. If `props.conf` already exists, modify it by adding the content shown next. (Writing the `props.conf` file is covered in Chapter 11.)

Props.conf

```
[Test9]
```

```
TIME_PREFIX=\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\s\-\s\d{5}\s+
```

```
TIME_FORMAT = %m/%d/%Y %k:%M
```

```
MAX_TIMESTAMP_LOOKAHEAD = 15
```

```
LINE_BREAKER = ([\r\n]+)\d+\s+\\"$EIT\,
```

```
SHOULD_LINEMERGE = false
```

```
TRUNCATE = 99999
```

4. Download the `test.txt` file from <https://github.com/deppmehta/splunk-certification-guide/blob/main/ch1/Test.txt>
5. Click Add Data, as shown in Figure 1-16.

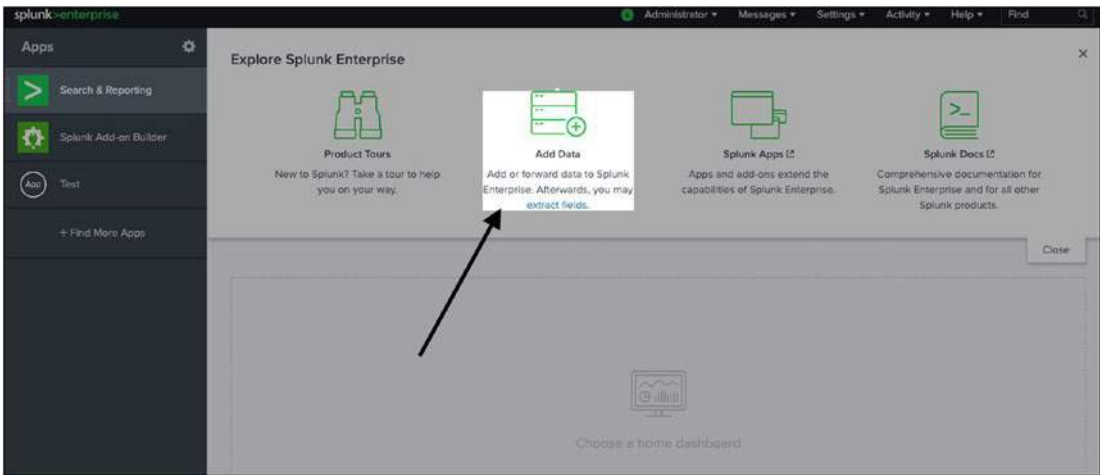


Figure 1-16. Add Data: Splunk Web

6. Click Upload, as shown in Figure 1-17.

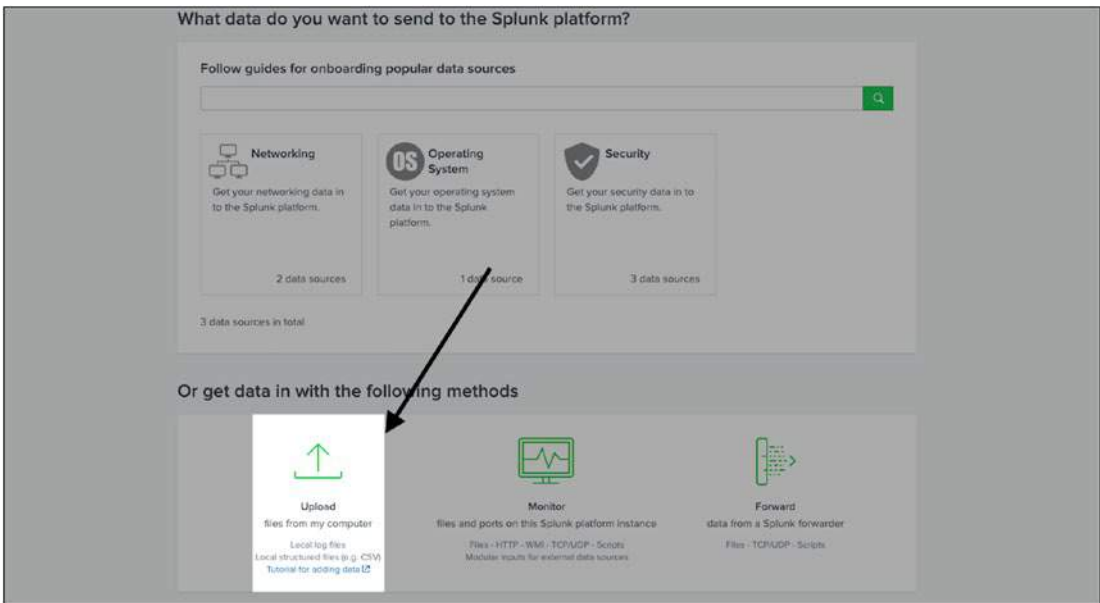


Figure 1-17. Upload Data: Splunk Web

7. Click the Source file, and then click Next.
8. In the Set Source Type screen, note that the current time is displayed rather than when the events occurred. For a better understanding, look at Figure 1-18.

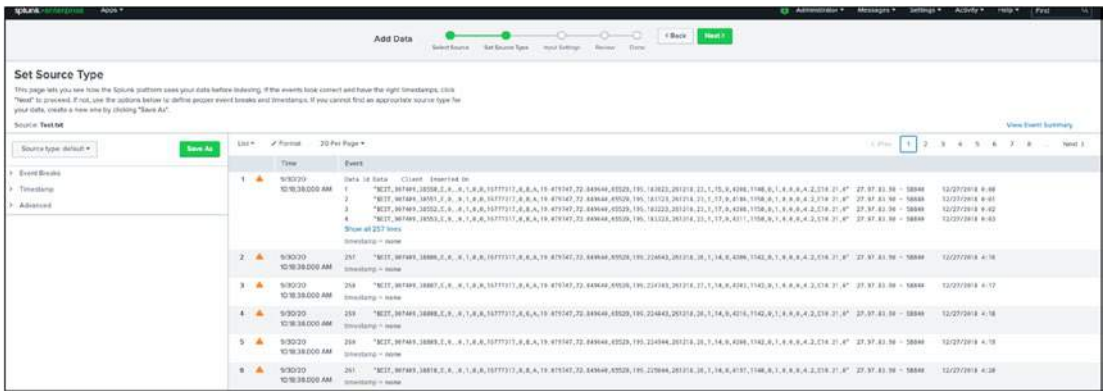


Figure 1-18. test.txt Events: Improper Timestamp

- In the Source Type box, enter **test9** and select Test9 as the source type. In the event breaks, select every line for incoming data. Now, the data can be transformed. The timestamp is selected from the Time field already present in your event, which is due to the props.conf file that you edited. Figure 1-19 shows the Source Type box.

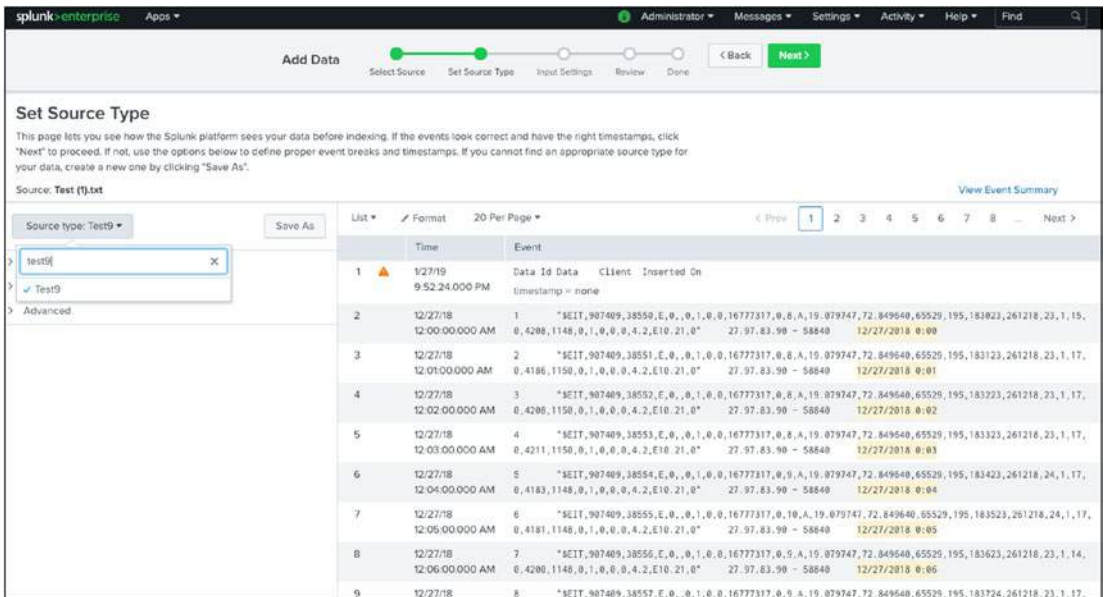


Figure 1-19. test.txt Events: Extracted Timestamp

10. Create a new index, called Test, in the Input settings. The Test index stores all the events from the test.txt file. Figure 1-20 shows how to create an indexer. You only need to enter the name.

**New Index** [X]

**General Settings**

Index Name:   
Set index name (e.g., INDEX\_NAME). Search using index=INDEX\_NAME.

Index Data Type:  Events  Metrics  
The type of data to store (event-based or metrics).

Home Path:   
Hot/warm db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/db).

Cold Path:   
Cold db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/colddb).

Thawed Path:   
Thawed/resurrected db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/thaweddb).

Data Integrity Check:  Enable  Disable  
Enable this if you want Splunk to compute hashes on every slice of your data for the purpose of data integrity.

Max Size of Entire Index:  GB ▾  
Maximum target size of entire index.

Max Size of Hot/Warm/Cold Bucket:  GB ▾

[Cancel] [Save]

**Figure 1-20.** Test Index: Splunk Web

11. Click the Save button after you look at all the fields.
12. Go to the test app and enter the **index="Test"** Splunk processing command in the search box. In the time bar, which is located next to the search box, select the All Time range. You see all the events in Splunk.

You now have a better idea of Splunk software and saw how to input data. Now, you have successfully arrived at the end of Chapter 1.

## Summary

This chapter highlighted the underlying concepts of Splunk by discussing its history, inception, and salient features. It provided a roadmap for the admin exam and explained the foundational architecture and how to install Splunk on macOS and Windows. In the last section, you saw how to play with the data in Splunk, including data onboarding, modifying source files (carrying out activities such as adding timestamps), and line breakers.

You learn more about `.conf` files in Chapter 11. Until then, try to onboard the various types of data and play with them on Splunk.

In the next chapter, you learn about Splunk's Search Processing Language and its various commands.

## Multiple-Choice Questions

- A. Splunk Enterprise components can only be installed and administered on-premises.
  - 1. true
  - 2. false
- B. What is the name of the default app when you install Splunk for the first time?
  - 1. Home app
  - 2. Searching and Reporting
  - 3. Splunk DB app
  - 4. Splunkforwarder

- C. What are the phases of Splunk? (Select all that apply.)
1. parsing
  2. input
  3. licensing
  4. indexing
- D. Default metadata is added to Splunk when indexing your data source.
1. true
  2. false
- E. What is the default Splunk management port?
1. 8000
  2. 8089
  3. 8056
  4. 9997

**Answers**

- a. 1
- b. 1
- c. 1, 2, and 4
- d. 1
- e. 2

## Further Reading

More information on how indexing works is at <https://docs.splunk.com/Documentation/Splunk/7.2.3/Indexer/Howindexingworks>.

You can look at the architecture of Splunk here: <https://docs.splunk.com/File:Architecture-new.png>.

You can read about what Splunk does with your data at <https://docs.splunk.com/Documentation/Splunk/7.2.3/Data/WhatSplunkdoeswithyourdata>.

## CHAPTER 2

# Splunk Search Processing Language

In Chapter 1, you learned about Splunk's architecture, history, inception, and salient features. You saw a roadmap for the Splunk Enterprise Certified Admin exam and were introduced to Splunk in a nutshell. You installed Splunk on macOS or Windows and went through the process to add data to it. In this chapter, you take a deep dive into the Splunk Search Processing Language and the methods to analyze data using Splunk.

Splunk's Search Processing Language (SPL) is a user-friendly language developed to search data that has been indexed in Splunk. The language is based on Unix piping and Standard Query Language (SQL). The SPL's scope includes data searching, filtering, modification, manipulation, insertion, and deletion. You can group different events, get insight into data patterns, read real-time interactive reports, and gain business intelligence with this software. The SPL can perform various functions through a single command or a group of commands using a pipeline. Therefore, it is necessary to discuss this language before successfully operating Splunk. Chapter 3 also incorporates Splunk Search Processing Language examples.

This chapter discusses the following topics.

- The pipe operator
- Time modifiers
- Basic SPL
- Sorting results
- Filtering commands
- Reporting commands
- Filtering, modifying, and adding fields
- Grouping results

## The Pipe Operator

In SPL, the pipe operator executes chains of search commands in Splunk. It is represented with the `|` operator. All the commands are executed from left to right in SPL, where the command to the left of the pipe operator is executed first, followed by the command to its right. Simply put, the output to the left of the pipe operator is the input to the command to the right of the pipe operator.

1. Go to the test app in Splunk Web and enter the following command in the search bar.

```
index="Test"
```

2. To get the total event count in the Test index, use `index="Test"` to filter results and get all events of the Test index and use the pipe operator to pass the output of `index="Test"` as input to get the total counts of events in the Test index. Refer to the command in the search bar.

```
index="Test" | stats count
```

Field names in Splunk are case sensitive, but the field values are case insensitive. For example, in index Test [ `search index="Test" to="deep@gmail.com" | transaction sid` ]. Fields name `sid` is case sensitive but the values in `sid` are not case sensitive which results in one event `sid` matches another event `sid` value.

## Time Modifiers

When the index processes an event, the timestamp is saved as the default field `_time`. If you already have a timestamp in your raw data, you can update `_time` using `props.conf` (more about `props.conf` in Chapter 11). Whenever you execute an SPL command in Splunk, you need to be cautious about the time range. Once you fire an SPL query using a time range, only events that occurred during the time range are retrieved, the other events are filtered. The time range picker is located to the right of the search command.



There are two major types of time modifiers: relative search and real-time search.

- A **relative search** captures data that appears in a given amount of time, such as *yesterday*. The search refreshes when the relative time boundary has passed. So a search for *yesterday* refreshes (and is, therefore, relative to) each new day.
- A **real-time search** refreshes in real time; thus, it continually pushes old data and acquires new data, depending on the boundaries.

**Sorting results** in SPL orders results in ascending order or descending order. You can choose whichever order you want based on the `_time` parameter.

Time is the most efficient filter in SPL.

This section discussed the pipe operators that execute chain commands and time modifiers. In the next section, let's discuss SPL fundamentals.

## Understanding Basic SPL

Understanding the basics of SPL is like learning an alphabet to best understand Splunk. SPL consists of search language syntax, boolean operators, search modes, and syntax coloring.

## Search Language Syntax

Splunk has five basic search components.

- **Search terms** define the data you want to retrieve from Splunk. A term uses highlighted keywords. For example, the `index = "Test" 58840` processing command filters all events with the number 58840 incorporated in them.
- **Commands** define what to do with the retrieved results of a search. They are used mainly for analyzing data in a result set. For example, the `index="Test" |tail 20` processing command retrieves the last 20 results in the Test index.
- **Functions** define how to chart your results. For example, the `index="_internal"|stats avg(bytes)` processing command gives the average number of bytes in the `"_internal"` index.

- **Arguments** are the variables that you usually apply to functions to retrieve results. For example, the `index="Test"|eventstats sum(count) as total_count` processing command applies the count's summation function as the total count.
- **Clauses** let you group results and rename field results. For example, the `index="_internal"|stats avg(bytes) by the host` processing command gives the average byte used in the "\_internal" index and the average number of bytes used by each of the hosts.

Every search in Splunk is a job, and the default lifetime of that job is 10 minutes; it can be extended to up to 7 days.

## Boolean Operators in Splunk

There are three types of boolean operators in Splunk: AND, OR, and NOT. They always need to be capitalized.

The AND operator is implied between terms. Instead of writing an entire SPL statement again, you can use the boolean operator AND. For example, if you want to check for products that are available on both Amazon and Walmart, you can use the following query.

```
index="products" | product_website="amazon" AND "walmart"
```

The OR operator is applied to either of the terms. For instance,

```
index="products" | product_website="amazon" OR "walmart"
```

The NOT operator only applies to the term immediately following NOT, allowing you to omit certain results in a query. For example,

```
index="products" | error NOT(400 or 500)
```

What about != vs. the NOT operator? A search with the NOT operator checks if a field has a specified value, whereas a search with != retrieves events where fields exist, but the field does not need to have a value.

## Syntax Coloring in SPL

In Splunk, a part of the search string is automatically colored. Boolean operators, commands, arguments, and functions each have a different color; the defaults are shown in Table 2-1. Notably, you can customize or disable syntax coloring.

*Table 2-1. Syntax Colors in SPL*

Argument_Name	Color
boolean operator	orange
commands	blue
argument	green
functions	purple

The next section discusses sorting results.

## Sorting Results

Sorting results in SPL **order results in ascending order or descending order**. You can choose whichever order you want to appear based on the **\_time** parameter.

### Sort

Sorting is performed using the sort command. The field name follows the sort command. By default, the sort field name sorts the results in ascending order; to sort data in descending order, you must write **sort -fieldname**. Table 2-2 describes the various types of sort commands.

**Table 2-2.** *Sort Commands*

Command	Explanation
lsort field1	Sorts results in ascending order by field1
lsort count field1	Retrieves count the number of events in ascending order by field1
lsort -field1	Sorts results in descending order by field1
lsort count -field1	Retrieves the number of events in descending order by field1

To limit the sorted result, you can use the limit option. The following is an example syntax query that shows how to limit the sorted result.

```
"your_query | sort - sum(field_name)| head 10(count)"
```

That is all for sorting. In the following section, you learn various Splunk filtering commands.

## Filtering Commands

Filtering commands in SPL filter events based on various SPL commands, such as where, dedup, head, and tail. Let's start with the where command.

### where

The where filtering command evaluates SPL to filter the results. If the result matches, then the evaluation is successful, and the result is retrieved. If the result evaluation does not match, it is unsuccessful, and the result is not retrieved. Table 2-3 explains how to use the where clause to filter results.

**Table 2-3.** *The where Command*

Command	Explanation
<code>lwhere field1!=value</code>	A where statement checks if data in field 1 is not equal to field1. If so, it retrieves similar events that are not identical to field 1.
<code>lwhere field1=value</code>	A where statement checks if data in field 1 is the same; if it is, it retrieves similar events.
<code>lwhere field1&gt;value</code>	A where statement checks if data in field 1 is greater; if it is, it retrieves similar events.
<code>lwhere field1&lt;value</code>	A where statement checks if data in field 1 is less; if so, it retrieves similar events.
<code>l where (ip LIKE "27.%.%.%")</code>	The where clause tries to find all the events that have an IP range from 27.0.0.0 to 27.255.255.255.

## dedup

The dedup command removes all duplicate data that falls within the same criteria. The result matches are unique to the field value provided. There won't be any other duplicate data that has an equal value to the field value.

Table 2-4 describes the various dedup commands.

**Table 2-4.** *dedup Commands*

Command	Explanation
<code>ldedup field1</code>	dedup checks the field1 value for any other event with the same value; if so, it removes it.
<code>ldedup count field1</code>	dedup checks field1 value for any other event with the same value; if so, it removes the values. The dedup count determines the field1 count.

## head

The head command retrieves initial count events. Count is a parameter that stops retrieving results based on several counts provided to the head command; for example, `head 5` retrieves the first five events.

## tail

The `tail` command retrieves the last count events. Count is a parameter that stops retrieving results based on the count number provided to `tail`; for example, `tail 5` retrieves the last five events.

## Reporting Commands

The reporting command in SPL prepares a summary that is useful for reporting. These commands include `top`, `rare`, `history`, `stats`, `untable`, `timechart`, `chart`, and `table`.

## top

The `top` command retrieves a table with the top value (most common) in the field values, including the field's total count and the percentage. Table 2-5 explains how to use the `top` command for grouping events.

**Table 2-5.** *The top Command*

Command	Explanation
<code>!top field</code>	The <code>top</code> command finds the top (most common) value of all field values and retrieves the field1 table, a total count, and percentage.
<code>!top count field1</code>	The <code>top</code> command finds the top (most common) value of the field and then retrieves the count and creates a field1 table, a total count, and percentage.

By default, the output of the `top` command is in table format.

## rare

The **rare command** is the opposite of the **top command** because it shows the total number of times the rare values appear in the field and their percentage of the results. Table 2-6 explains how to use the `rare` command for grouping events.

**Table 2-6.** *The rare Command*

Command	Explanation
lrare field	This rare command finds the rare value of field values and retrieves a table, total count, and the percentage of field1.
lrare count field	This rare command finds out the rare values of all field values and retrieves the count, creates a table, a total count, and the percentage of field1.

## history

The history command in SPL is used to view the current user's search history.

Table 2-7 explains the history commands.

**Table 2-7.** *History Commands*

Command	Explanation
lhistory	Returns a table of search history
lhistory events=true	Returns all events in the search history

## table

The table command in SPL generates a table on all field names that you want to include in your report.

Table 2-8 shows the table command syntax.

**Table 2-8.** *The table Command Syntax*

Command	Explanation
ltable fieldname1,fieldname2	Creates a table on the only field name provided

## stats

The stats command calculates aggregate statistics, such as the average, count, and sum of the results. It is like SQL aggregation.

The first in line is the aggregate function.

## Aggregate Functions

Aggregate functions summarize the values of each event to create a single meaningful value. Table 2-9 describes various aggregate functions in Splunk.

**Table 2-9.** *Aggregate Functions*

Function Name	Command	Explanation
avg(field)	lstats avg(field_name)	Stats Avg function returns the average number of events for a field
count(field)	lstats count(field_name)	Stats count function returns a count of events for field name)
distinct_count(field)	lstats dc(field_name)	Stats distinct_count function returns distinct values for field
max(field)	lstats max(field_name)	Stats max function returns max value for field
median(field)	lstats median(field_name)	Stats median function returns middle-most value for field
min(field)	lstats min (field_name)	Stats min function returns min value for field
mode(field)	lstats mode(field_name)	Stats mode function returns frequent value of a field name)
sum(field)	lstats sum(field_name)	Stats sum function returns the sum of the field value
var(field)	lstats mode(field_name)	Stats var function returns the variance of the field value



## Event Order Functions

The event order function returns events in chronological or timestamp order. Table 2-10 explains various event order functions.

**Table 2-10.** *Event Order Functions*

Function Name	Command	Explanation
first(field)	lstats first(field_name)	Stats first function returns first value of field
last(field)	lstats last(field_name)	Stats last function returns last value of field

## Multivalue stats and chart Functions

The multivalue stats and chart functions return the value of field X as a multivalue entry. Table 2-11 explains various multivalue stats and chart functions in Splunk.

**Table 2-11.** *Multivalue stats and chart Functions*

Function Name	Command	Explanation
list(field)	lstats list(field_name)	This stats list function returns a list of field values
values(field)	lstats values(field_name)	The stats values function returns a list of distinct field values

## Timechart Functions

A time chart is a statistical aggregation applied to a field to produce a chart, with time used as the X axis. Table 2-12 explains the various timechart functions in Splunk.

**Table 2-12.** *timechart Functions*

Function Name	Command	Explanation
per_day(field)	lstats per_day(field_name)	Stats per_day returns the rate of field per day.
per_hour(field)	lstats per_hour(field_name)	Stats per_hour returns the of field per hour.
per_minute(field)	lstats per_day(field_name)	Stats per_minute returns the rate of field per minute.
per_second(field)	lstats per_second(field_name)	Stats per_second returns the rate of field per second.

## unable

The unable command converts results from a tabular format into a format similar to the statistics tab.

Table 2-13 describes the unable command.

**Table 2-13.** *The unable Command*

Command	Explanation
lunable fieldnameX fieldnameY fieldvalueY	They return duplicate events on Fieldname, but fieldvalue Y has unique values from fieldname X.

## chart

The chart command is a transforming command because it returns the results in the form of a table. This command can also display data in the form of a chart. You decide what is on the X axis.

By using the chart command, you can subgroup data by using them over and the by clause.

Table 2-14 shows various chart commands and their applications.

**Table 2-14.** *chart Commands*

Command	Explanation
<code>lchart count over field name</code>	It counts the events and retrieves them in the table format. You can plot result in the graph using visualization
<code>lchart count over filename by hostname</code>	This subgroups data as you have used them over and the by clause.
<code>lchart avg(fieldname) over host</code>	In this command, you use the avg function over the hostname.
<code>lchart avg(fieldname) over host by time</code>	In this command, you use the avg function over hostname by time as a subgroup clause.

## timechart

The timechart command is a transforming command. A timechart is a statistical aggregation command applied to the field at the Y axis to produce a chart, with time as the X axis.

Table 2-15 describes timechart commands.

**Table 2-15.** *timechart Commands*

Command	Explanation
<code>timechart count</code>	It counts the number of events at a particular time based on the data source provided.
<code>timechart count by fieldname</code>	It counts the number of events at a particular time based on the field name provided.

At this point in the chapter, you are done with the pipe operators, time modifiers, the search language syntax, and the three boolean commands.

## Filtering, Modifying, and Adding Fields

This section analyzes filtering, modifying, and adding fields in a report. A few the commands include eval, rex, lookup, and field.

### eval

The eval command calculates the expression and puts the resulting value into search field values. If (x1, y1, x2, y2) is evaluated, then the first expression, x1, is a boolean expression. If it returns True, a result of y1 is evaluated. If it is False, the next boolean expression, x2, is evaluated.

There are several functions to discuss.

### Comparison and Conditional Functions

Comparison and conditional functions compare values or specify conditional statements. Table 2-16 explains the various comparison and conditional functions in Splunk.

**Table 2-16.** Comparison and Conditional Functions

Function Name	Command	Explanation
case(x,"Y",x1,"Y1"	eval Description=case(fieldname<= value, "message", fieldname>value AND fieldname<=value, "message", fieldname>value, "message")	In this statement it accepts alternating condition
Cidrmatch("X",Y)	eval isLocal=if(cidrmatch("value ",Fieldname), "Message", "Message") table isLocal	Returns True or False based on matches of CIDR
coalesce(X,...)	eval ip=coalesce(fieldname1, fieldname2)	It evaluates X if True, return Y; otherwise, it returns Z
if(X,Y,Z)	eval err=if(fieldname == value, "message", "message")	It evaluates X, if True, return Y; otherwise it returns Z

*(continued)*

**Table 2-16.** (continued)

Function Name	Command	Explanation
in(FIELD,VALUE-LIST)	where fieldname in("value", "value", "value", "value")	It returns True, if one of the fields is validated.
Like(TEXT,PATTERN)	eval is_a_foo=if(like(field, "value"), "message", "message")	It returns True if text matches the pattern.
match(SUBJECT,"REGEX")	eval k = if(match(field_value,"message"), 1, 0)	It returns True or False based on regex matching subject
null()	eval n=nullif(fieldA,fieldB)	It takes no arguments and returns NULL.
nullif(X,Y)	eval n=nullif(fieldA,fieldB)	It compares the field. If X and Y are the same, it returns NULL; otherwise, it returns X.
Searchmatch(X)is	makeresults 1   eval _raw = "x=message1 y=message2"   eval x="message1"   eval y="message2"   eval test=if(searchmatch("x=message1 y=*"), "yes", "no")   table _raw test x y	It returns True if searchmatch(x) matches the event; otherwise, it returns False.
validate(X,Y,Z)	eval n=validate(isint(port), "Message", port >= range AND port <= range, "Message")	It is opposite the case function.

## Conversion Functions

The conversion function converts numbers into strings and strings into numbers. Table 2-17 explains various conversion functions in Splunk.

**Table 2-17.** *Conversion Functions*

Function Name	Command	Explanation
printf("formats",arguments)	eval new_field=printf("%04.4f %-30s",field_one,field_two)	It is similar to the sprintf() function in C.
tonumber(NUMSTR,BASE)	eval n=tonumber(fieldname)	It converts input string NUMSTR to a fieldname.
tostring(X,Y)	eval foo=615   eval foo2 = tostring(foo, "Message")	It converts the input value into a string.

## Cryptographic Functions

Cryptographic functions in Splunk compute the secure hash of string values. Table 2-18 explains various cryptographic functions in Splunk.

**Table 2-18.** *Cryptographic Functions*

Function Name	Command	Explanation
md5(X)	eval n=md5(field)	It computes and returns the MD5 hash of X.
sha1(X)	eval n=sha1(field)	It computes and returns the secure hash of String X based on SHA-1 hash function.
Sha256(X)	eval n=sha256(field)	It computes and returns the secure hash of String X based on SHA-256 hash function.
sha512(X)	eval n=sha512(field)	It computes and returns the secure hash of String X based on the SHA-512 hash function.

## Date and Time Functions

Date and time functions in Splunk contain the functions that can **calculate dates and times**. Table 2-19 describes the various date and time functions in Splunk.

**Table 2-19.** *Date and Time Functions*

Function Name	Command	Explanation
now()	leval k=now()	It takes no argument and returns time when the search is started.
relative_time(X,Y)	leval k=relative_time(now(),"-15d@d")	It takes X as first argument and Y as second argument and displays all events inside of the range.
strftime(X,Y)	eval min_sec=strftime(_time, "%M:%S")	It takes X as a Unix time value as first argument and Y as a String time value.

## Informational Functions

The informational functions contain a certain command that returns the information about values. Table 2-20 explains the informational functions in Splunk.

**Table 2-20.** *Informational Functions*

Function Name	Command	Explanation
isint(X)	eval n=if(isint(field_name),"int", "not int") table n	It takes one argument and returns True if the argument is True; otherwise, it returns False.
isnotnull(X)	eval n=if(isnotnull(field_name),"yes", "no") table n	It takes one argument and returns yes or no based on the argument if it's null or not null.
isnull(X)	eval n=if(isnull(field_name),"yes", "no") table n	It takes one argument and returns yes or no based on the argument if it's null or not null.
isnum(X)	eval n=if(isnum(field_name),"yes", "no") table n	It takes one argument and returns yes or no based on the argument value. It returns yes if the argument is a number.
isstr(X)	eval n=if(isstr(field_name),"yes", "no") table n	It takes one argument and returns yes or no based on argument value. It returns yes if the argument is a string.
typeof(X)	eval n=typeof(12) + typeof("string") + typeof(1==2) + typeof(badfield) table n	It takes one argument and returns the data type of the argument.

## Mathematical Functions

Mathematical functions comprise functions that can perform mathematical calculations. Table 2-21 explains the various mathematical functions in Splunk.

**Table 2-21.** *Mathematical Functions*

Function Name	Command	Explanation
abs(X)	<code>!makeresults   eval a=abs(number)</code>	It takes number X and it returns its absolute value.
ceiling(X)	<code>!makeresults   eval n=ceil(1,9)</code>	It rounds the number X to the next highest integer.
exp(X)	<code>!makeresults   eval y=exp(3)</code>	It returns number and exponential value of a function.
floor(X)	<code>!makeresults   eval n=floor(1.9)</code>	It rounds a number X down to the nearest whole integer.
ln(X)	<code>!makeresults   eval lnBytes=ln(bytes)</code>	It takes number X and returns its natural logarithm.
log(X,Y)	<code>!makeresults   eval num=log(number,2)</code>	It takes either one or two numeric arguments and returns the logarithm of the first argument X using the second argument Y as a base.
pi()	<code>!makeresults   eval area_circle=pi()*pow(radius,2)</code>	It takes no arguments and returns the constant pi to 11 digits of precision.
sqrt(X)	<code>!makeresults   eval n=sqrt(9)</code>	It takes one argument and returns its square root

## Multivalue eval Functions

The multivalue eval function returns multivalue fields using commands such as `mvappend` and `mvdedup`. Table 2-22 explains the types of multivalue eval functions.



**Table 2-22.** *Multivalue Eval Functions*

Function Name	Command	Explanation
commands(X)	eval x=commands("")	It takes a search string, or field that contains a search string, such as X and returns a multivalued field containing a list of the commands used in X.
mvappend(X,...)	eval fullName=mvappend(initial_ values, "middle value", last_values)	It takes N number of arguments and returns a result of all the values. It can be strings, multivalue fields, or single value fields.
mvdedup(X)	eval s=mvdedup(mvfield)	It takes a multivalue field and returns a multivalued field with duplicate values removed
mvsort(X)	eval s=mvsort(mvfield)	It takes a multivalued field and returns a field sorted lexicographically.

## Statistical eval Functions

Statistical eval functions are evaluation functions to calculate statistics. Commands such as max, min, and random are types of statistical eval functions.

Table 2-23 explains these functions and their respective commands.

**Table 2-23.** *Statistical Eval Functions*

Function Name	Command	Explanation
max(X,Y,..)	eval n=max(1, 3, 6, 7, "foo", field)	It takes an arbitrary input and returns the maximum result.
min(X,...)	eval n=min(1, 3, 6, 7, "foo", field)	It takes an arbitrary input and returns the minimum result.
random()	makeresults   eval n=random()	It takes no arguments and returns a pseudo-random integer ranging from zero to 231-1.

## Text Functions

Text functions return information about strings and numeric fields in functions and nesting functions. `len`, `lower`, `ltrim`, and `rtrim` are among the types of text functions in Splunk.

Table 2-24 explains these functions and their commands.

**Table 2-24.** *Text Functions*

Function Name	Command	Explanation
<code>len(X)</code>	<code>leval k= len(field_name) table k</code>	It returns the <code>field_name</code> count.
<code>lower(X)</code>	<code>leval k= lower(field_name) table k</code>	It takes one field value and returns string to lowercase.
<code>ltrim(X,Y)</code>	<code>leval k= ltrim(Field_name,"value") table k</code>	It takes one or two arguments, X and Y, and returns X with the characters in Y trimmed from the left side. If Y is not specified, spaces and tabs are removed.
<code>rtrim(X,Y)</code>	<code>leval k= rtrim(Field_name,"value") table k</code>	It takes one or two arguments, X and Y, and returns X with the characters in Y trimmed from the right side. If Y is not specified, spaces and tabs are removed.
<code>upper(X)</code>	<code>leval k= upper(field_name) table k</code>	It takes one field value and returns string to the uppercase.

## Trigonometric and Hyperbolic Functions

To calculate trigonometry and hyperbolic values, the Splunk contains commands like `acos`, `acosh`, `asin`, and `asinh`. They are explained in Table 2-25.

**Table 2-25.** *Trigonometry and Hyperbolic Values*

Function Name	Command	Explanation
acos(X)	eval n=acos(0)	It computes the arc cosine of X in the interval [0,pi] radians.
acosh(X)	eval n=acosh(2)	It computes the arc hyperbolic cosine of X in radians.
asin(X)	eval n=asin(1)	It computes the arc sine of X in the interval [-pi/2,+pi/2] radians.
asinh(X)	eval n=asinh(1)	It computes the arc hyperbolic sine of X in radians.
sin(X)	eval n=sin(1)	It computes the sine of X.
sinh(X)	eval n=sinh(1)	It computes the hyperbolic sine of X.
tan(X)	eval n=tan(1)	It computes the tangent of X.
tanh(X)	eval n=tanh(1)	It computes the hyperbolic tangent of X.

## Rex

The rex command extracts fields from regular expression-named groups or replaces characters in a particular field using the sed expression.

The rex command matches the value specified against the expression. Table 2-26 explains the rex command.

**Table 2-26.** *The rex Command*

Command	Explanation
rex field=_raw "From: <(?!<from>.*>> To: <(?!<to>.*>>"	It extracts the field values; for example, "from" and "TO " are extracted.

Mode as Sed replaces the substitute character to the given value of the field. This feature is explained as follows.

```
| rex field=fieldname mode=sed "value"
```

It replaces the current value of the field with the value you specified.

## lookup

The lookup command enriches user data by adding field value combinations from tables. Lookup in SPL adds fields to event data in results obtained from the search.

Table 2-27 explains the lookup function.

**Table 2-27.** *The lookup Function*

Command	Explanation
lookup lookup_name Fieldname OUTPUTNEW field_name2	In this command, lookup_name tries to match a field's value in every event and return field_name2.

There are two types of lookups: input and output.

### Input Lookup

An input lookup searches the contents of a lookup table. (More on this in Chapter 4.)

The commands of input lookup are explained in Table 2-28.

**Table 2-28.** *Input Lookup*

Command	Explanation
inputlookup test	This command reads from the test lookup that is defined in transforms.conf.
inputlookup append=t test	This command reads from the test lookup defined in transforms.conf and appends the field to the count result.

### Output Lookup

The output lookup writes fields in search results to a static lookup table file.

The output lookup commands are explained in Table 2-29.

**Table 2-29.** *Output Lookup Commands*

Command	Explanation
outputlookup lookupname	This command writes to look up the name in the lookup table as specified in transforms.conf.
outputlookup test.csv	This command writes to the test.csv lookup file under \$SPLUNK_HOME/etc/system/lookups or \$SPLUNK_HOME/etc/apps/*/lookups.

Simply put, lookup enriches the field value data through combinations.

## Field

The field command keeps or removes search field results based on the conditions provided (see Table 2-30).

**Table 2-30.** *The field Command*

Command	Explanation
fields -fieldname1,fieldname2	Removes fieldname1,fieldname2 from the result.
fields fieldname1,fieldname2   fields - _*	It only keeps fieldname1,fieldname2 and removes the rest of the field from the result.

The last section of this chapter discusses the grouping results.

## Grouping Results

Grouping results in SPL determine the total number of events in a specified `_time` range for a particular ID, as well as for the field that is required. It is useful to recognize the patterns from the events

The command for grouping is the transaction command.

## Transaction

The transaction command is used when you require all your events to be correlated and must define event grouping based on the start and end value. The maximum default value of a transaction is 1000.

Table 2-31 describes the various transaction commands for grouping results.

**Table 2-31.** *The Transaction Command*

Command	Explanation
<code>!transaction fieldname</code>	It displays all transactional events with the fieldname as specified
<code>!transaction fieldname maxpause=value</code>	It displays all transactional events with fieldname as specified having a maximum pause between events =value given
<code>!transaction fieldname maxspan=value</code>	It displays all transactional events with fieldname as specified having maximum span between events =value given
<code>!transaction fieldname maxspan=value and maxpause=value</code>	It displays all transactional events with fieldname, having a maximum span between events=value given and maximum pause=value given

## Summary

This chapter gave an overview of Splunk SPL. You learned about the search pipeline and the search processing language sorting command. You became familiar with the syntax coloring and result sorting features and the boolean operators AND, OR, and NOT. You also learned the filtering and reporting commands and the fields in Splunk. To learn more about these topics, please refer to <https://docs.splunk.com/Documentation/Splunk/7.2.4/SearchReference/ListOfSearchCommands>.

The next chapter discusses field extraction, macros, and field aliases and presents an example SPL query.

## Multiple-Choice Questions

- A. Time is the most efficient filter in Splunk SPL.
  - 1. true
  - 2. false
- B. What is the default timeline for any search job in Splunk?
  - 1. 10 minutes
  - 2. 7 minutes
  - 3. 7 days
  - 4. 10 days
- C. By default, the output of the top command is in table format.
  - 1. false
  - 2. true
- D. In the stats command what do DC functions perform?
  - 1. returns the number of events that match the search criteria
  - 2. returns a count of unique values for a given field
  - 3. returns the sum of a numeric field
  - 4. lists all the values in a field
- E. Filter, modify, and add fields in SPL filters and add \_\_ in a report.
  - 1. fields
  - 2. key
  - 3. value
  - 4. none of the above
- F. Select all the functions of the stats command.
  - 1. count
  - 2. distinct\_count

3. sum
  4. chart
  5. values
  6. list
- G. The timechart function is a part of \_\_\_\_
1. sorting results
  2. filtering commands
  3. reporting commands
  4. grouping results

**Answers**

- a. 1
- b. 1
- c. 2
- d. 2
- e. 1
- f. 1, 2, 3, 5, 6
- g. 3

## References

If you want to further explore Splunk searches, refer to the following reference materials.

- <https://docs.splunk.com/Documentation/Splunk/7.2.4/SearchReference/ListOfSearchCommands>
- <https://docs.splunk.com/Documentation/Splunk/7.2.4/Search/Specifytimemodifiersinyoursearch>
- <https://docs.splunk.com/Documentation/Splunk/7.2.4/SearchReference/WhatsInThisManual>



## CHAPTER 3

# Macros, Field Extraction, and Field Aliases

Chapter 2 was an overview of the Splunk Search Processing Language (SPL), time modifiers, and the pipeline operator. This chapter discusses field extraction, macros, and field aliases in Splunk and explores SPL by using various queries. Field extraction in Splunk is a process that extracts fields from raw data. Splunk can extract data fields during indexing and searching. Macros in Splunk are a reusable block (content that can be saved for future use) in which you can dynamically set the same logic for different parts or values in the dataset. Macros are useful when you want to frequently run the search command because it saves you from rewriting the whole command. Field aliases in Splunk provide fields alternate names. Field aliases correlate with various events that have similar field values.

Splunk search queries offer flexibility when working with small and big data. An SPL query helps you understand how to work with SPL and become familiar with SPL syntax.

This chapter covers the following topics.

- Field extraction
- Macros
- Field aliases
- An example SPL query

At the end of this chapter, you will be able to do a field extraction in Splunk using regular expressions and delimiters, create macros and field aliases, and fire a query in SPL. In other words, you will have covered up to 30% of the Splunk Power User exam blueprint.

Let's begin with field extraction.

## Field Extraction in Splunk

Field extraction is about creating new fields. Splunk extracts default fields for each event it indexes, such as host, source, and source type. Field extraction extracts data fields during indexing and searching. It provides two methods to extract data: regular expressions and delimiters.

### Regular Expressions

Regular expressions are mainly used for **unstructured** event data. Regular expressions are patterns used to match character combinations in string. In Splunk Web you just need to select the Regular Expression field and the value, it automatically generates regular expressions that match fields in other events. If the regular expression is unable to map the fields automatically you can also manually edit the expression and discard the event accordingly.

In Chapter 1, you onboarded data in Splunk. You can use regular expressions on that data to find IPs (Internet Protocols) and ports.

The various methods for using regular expressions are explained next.

### Regular Expression Using Field Extraction

In this section, you use field extraction to extract a port's field from index = "Test". The events whose values don't match the port's value are removed using a blacklist.

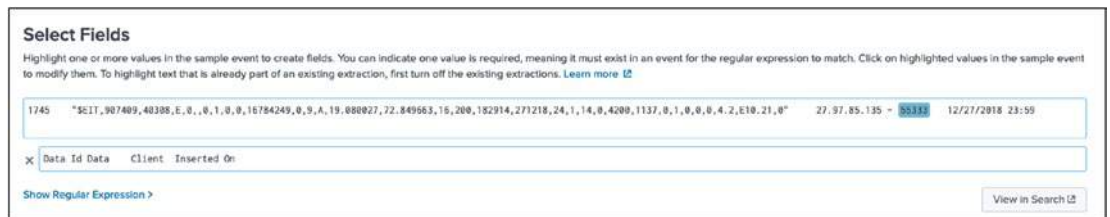
The following steps extract a port's value using regular expressions.

1. Click Splunk and go to Enterprise at the top of your screen.
2. Go to the Test app created in Chapter 1.
3. Search events for index="Test" and select All Time as the time range.
4. Expand the event and click Extract Fields.
5. Click Regular Expression and then click Next.
6. Select the port number from the event, as shown in Figure 3-1. In the Field Name, type **ports**.



**Figure 3-1.** Regular Expression Field Extraction

7. Click Next, and you see all events, matches, and non-matches.
8. Select the non-match event. It is blacklisted (a blacklist is a filtering rule that excludes an event from a set), and you find your regular expression. The Select Fields screen is similar to the one in Figure 3-2.



**Figure 3-2.** Regular Expressions Blacklisting

9. Click Next and then Save.

Next, let's look at inline regular expressions, which blacklist events values that don't match the port's value.

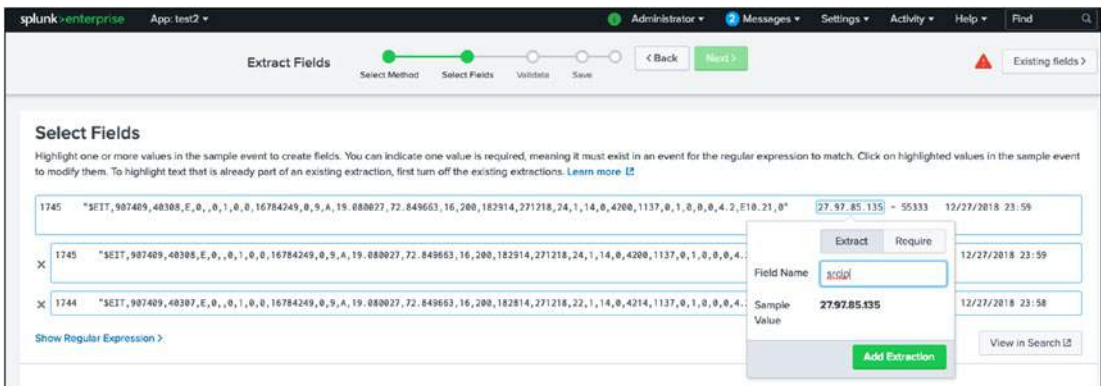
## Inline Regular Expression Using Field Extraction

In this section, you use inline regular expressions to extract ports field from index = "Test" and blacklist all events values whose values don't match the port's value.

The following steps extract ports in inline regular expressions.

1. Click Splunk and go to Enterprise at the top of your screen.
2. Go to the Test app and search events for index="Test", and then select All Time as the time range.
3. Expand the event and click Extract Fields.
4. Click Regular Expression, and then click Next.
5. Select the IP address.
6. In the Field Name, type **scrip**.
7. Click the Require button and select a few events.

A sample Select Fields screenshot is shown in Figure 3-3.



**Figure 3-3.** *Inline Regular Expressions*

8. Click Next, and then click Show Regular Expression, and then click Edit it.
9. Replace the regular expression content with the following.  

$$(?[^\-]*(:|-|\-.*-))^(?:[^\t\n]*\t){2}(?P<ip>[^\ ]+)$$
10. Remove the blacklisted content as done in the previous example, click Next, and then click Please Save.
11. Click Next, and save the report name as Test1.

Once the regular expression is edited, you cannot go back to Field Extractor UI. Next, let's discuss delimiters, another field extraction method.

## Delimiters

Delimiters are used in structured data. The fields are separated by a common delimiter, such as a comma, a space, a tab, or so forth. Delimiters are used in CSV data and similar formats.

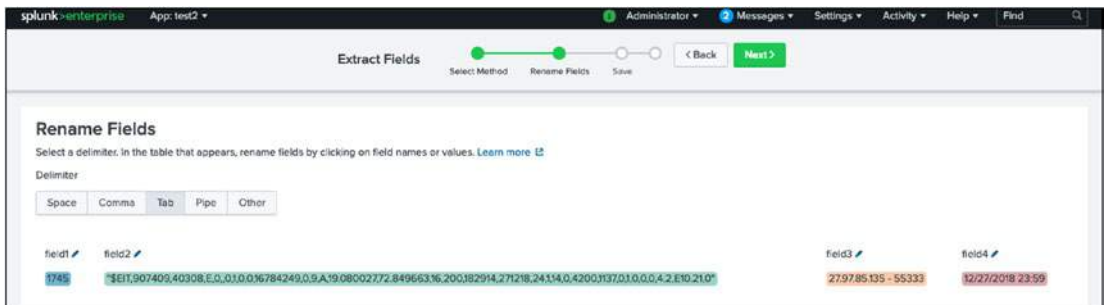
Delimiters are used only through field extraction.

## Delimiters Using Field Extraction

In this section, you use delimiters to extract fields from index="Test" and use tabs to separate fields from one another.

The following are the steps to use Delimiters.

1. Click Splunk, and go to Enterprise at the top of your screen.
2. Go to the Test app and search events for index="Test" and select All Time as the time range.
3. Expand the event and click Extract Fields.
4. Click Delimiters, and then click Next.
5. Separate the delimiter by selecting Tab. Figure 3-4 shows the Delimiter screen.



**Figure 3-4.** *Delimiters Fields Extraction*

6. Rename all fields as follows.
  - field1=id
  - field2=lat\_long
  - field3=ip\_port
  - field4=timestamp

Figure 3-5 shows the renaming process.



**Figure 3-5.** *Renaming Fields In Delimiters*

7. Click Next and save Report name as Test2.
8. Click Finish.

Use delimiters when your event contains structured data, such as JSON files, CSV files, and so forth.

Let’s discuss search command macros in the next section.

## Macros

Macros in Splunk can be a full search command or part of the search command in SPL. Macros are useful when you want to frequently run the search command because you do not need to rewrite the whole command.

There are two ways to create macros in Splunk.

- Using Splunk Web
- Using a .conf file

Macros are reusable knowledge objects since multiple people and multiple apps can use them.

## Create a Macro Using Splunk Web

Macros in Splunk are a reusable block in which you can dynamically set the same logic on different parts or values in the data set. Use Splunk Web to create a macro for the test app and make a macro for a transaction in which the max pause is 20 minutes.

To create macros using Splunk Web, follow these steps.

1. Click Settings and go to Advanced Search.
2. Select Search Macros and click New Macro.
3. In Destination app, type test.
4. In Name, type **session**.
5. In Definition, type **transaction maxpause=20m**.
6. Click Save.

Figure 3-6 shows the Add New page where these steps are performed.

The screenshot shows the 'Add New Macro' form in Splunk Web. The form is titled 'Add New Macro' and is located in the 'test' app. The form fields are as follows:

- Destination app:** test
- Name:** session
- Definition:** transaction maxpause=20m
- Use eval-based definition?:**
- Arguments:** (empty)
- Validation Expression:** (empty)
- Validation Error Message:** (empty)

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

**Figure 3-6.** Create a Macro

- 7. To test the macro, go to the search bar and type the following command.

```
index="test"|`session`|timechart span=1d sum(eventcount)| as sessions
```

Figure 3-7 shows the output.

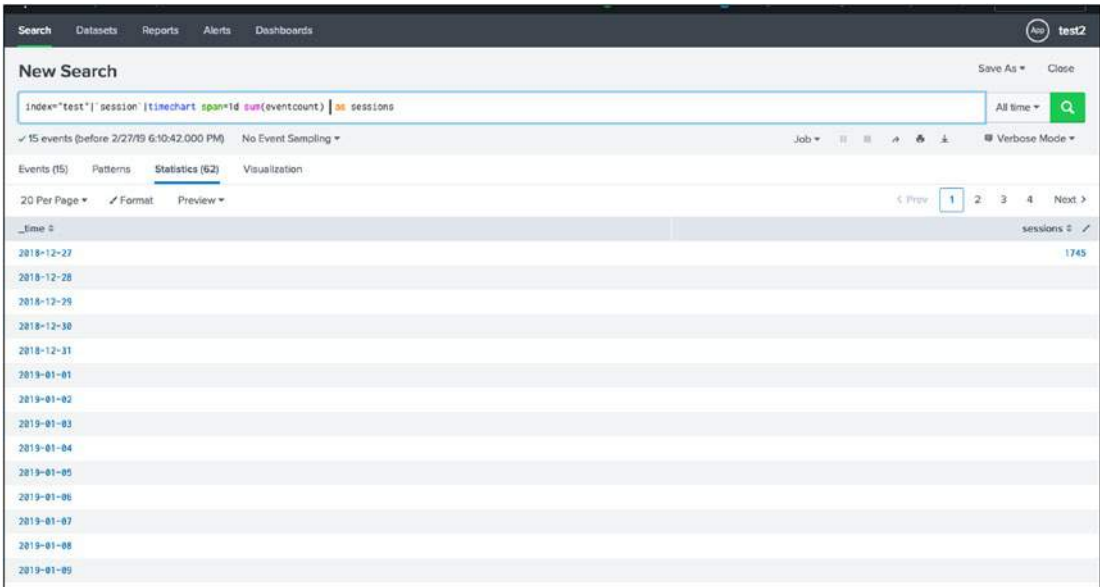


Figure 3-7. Macro's Output

---

**Note** To run macros in SPL, you must use ``macro _name``.

---

Next, let's look at another method for creating macros.

## Create a Macro Using the .conf File

To create a macro using the `.conf` file in Splunk, go to `$Splunk_Home/etc/apps/<app_name>/local/macros.conf`. If it is not there, create it; if it is there, edit it.

In this section, you create a macro for the test app using the `.conf` file. The macro would be a transaction with a max pause of 20 minutes. Go to `$Splunk_Home/etc/apps/Test/local/`. If `macros.conf` is there, edit it; otherwise, you need to create it.



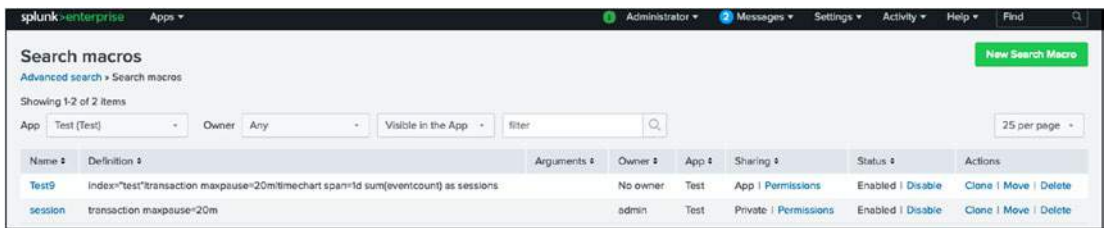
Once the `macros.conf` file is ready, follow these steps.

1. Add the following block.

```
[Test9]
definition = index="test"|transaction maxpause=20m|timechart
span=1d sum(eventcount) as sessions
```

2. Restart Splunk.
3. Go to Settings.
4. Go to Advanced Search.
5. Search macros.
6. Go to Test9 as an added macro using `.conf`.

The working Search Macros page is illustrated in Figure 3-8.

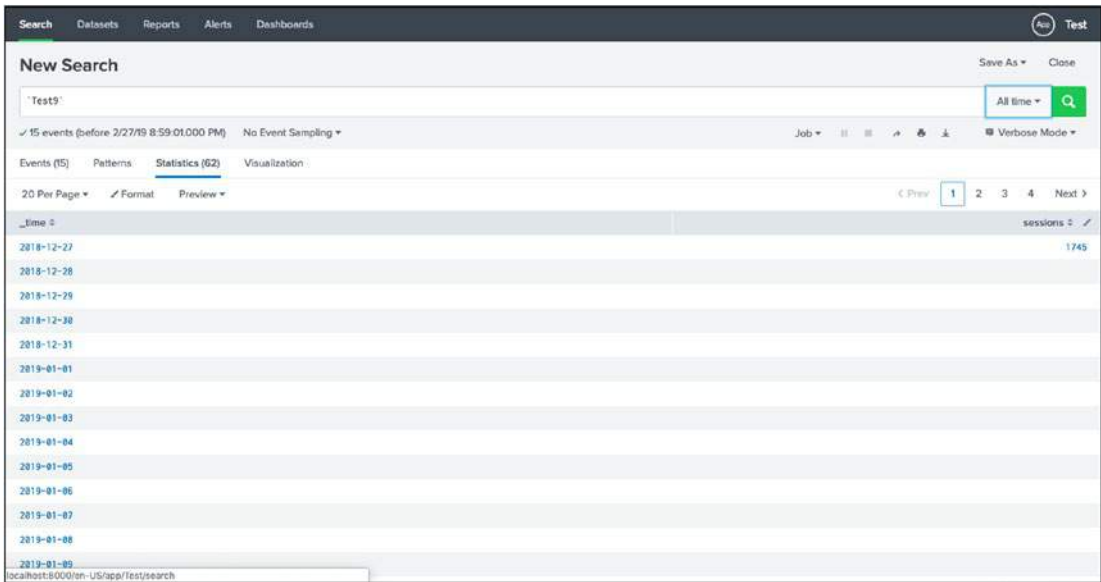


**Figure 3-8.** Illustrated Macros:test app

7. To test this macro, go to the search bar and type the following command.

```
index="test"|`session`|timechart span=1d sum(eventcount) as
sessions
```

Figure 3-9 shows the output.



**Figure 3-9.** macro Test9

Next, let’s discuss field aliases.

## Field Aliases in Splunk

Field aliases are for **correlating various events in Splunk** that have similar field values. To get greater insights into these events, you use field aliases to group two or more fields with similar field values.

Let’s look at setting up field aliases in Splunk environments.

## Setting up Field Aliases

In this section, you upload two data sources: Test2.txt and Test3.csv. In Test2.txt, you have “unit\_id” and in Test3.csv, you have “id”. You create unit\_id and id field aliases.

You can download Test2.txt from <https://github.com/deeppehmeta/splunk-certification-guide/blob/main/ch3/Test2.txt>.

To set up the field aliases, follow these steps.

1. Go to Add Data and upload Test2 .txt.
2. Set Source Type as default and Event Breaks as Every line.

Figure 3-10 shows the Set Source Type page.

**Set Source Type**

This page lets you see how the Splunk platform sees your data before indexing. If the events look correct and have the right timestamps, click "Next" to proceed. If not, use the options below to define proper event breaks and timestamps. If you cannot find an appropriate source type for your data, create a new one by clicking "Save As".

Source: **Test2.txt**

Source type: default Save As

Event Breaks

Define event boundaries for incoming data.

Event breaking Policy:  Auto  Every Line  Regex

Timestamp

Advanced

Time	Event
10/24/19 2:18:39.000 PM	unit_id method timestamp status Category location timestamp = none
3/24/19 12:00:00.000 AM	1 get 3/24/2019 8:00 200 sports 1
3/24/19 12:00:00.000 AM	2 get 3/24/2019 8:00 200 Games 2
3/24/19 12:00:00.000 AM	3 get 3/24/2019 8:00 200 Entertainment 3
3/24/19 12:00:00.000 AM	4 get 3/24/2019 8:00 200 sports 4
3/24/19 12:00:00.000 AM	5 get 3/24/2019 8:00 200 sports 5
3/24/19 12:00:00.000 AM	6 get 3/24/2019 8:00 200 sports 5
3/24/19 12:00:00.000 AM	7 get 3/24/2019 8:00 200 sports 7
3/24/19 12:00:00.000 AM	8 get 3/24/2019 8:00 200 sports 1
3/24/19 12:00:00.000 AM	9 get 3/24/2019 8:00 200 sports 1
3/24/19 12:00:00.000 AM	10 get 3/24/2019 8:00 200 sports 1
3/24/19 12:00:00.000 AM	11 get 3/24/2019 8:00 200 sports 5
3/24/19 12:00:00.000 AM	12 get 3/24/2019 8:00 200 sports 5
3/24/19 12:00:00.000 AM	13 get 3/24/2019 8:00 200 sports 3
3/24/19 12:00:00.000 AM	14 get 3/24/2019 8:00 200 sports 3

**Figure 3-10.** *Sourcetype:test89*

3. Click Next and create the new index as tes22, the app name as test, and the source type as test89.

You can download Test3.csv from <https://github.com/deepmehta/splunk-certification-guide/blob/main/ch3/Test3.csv>.

1. Go to Add Data and upload Data.
2. Select test3 .csv.
3. Set Source Type as default and Event Breaks as Every Line. The Set Source Type page is shown in Figure 3-11.

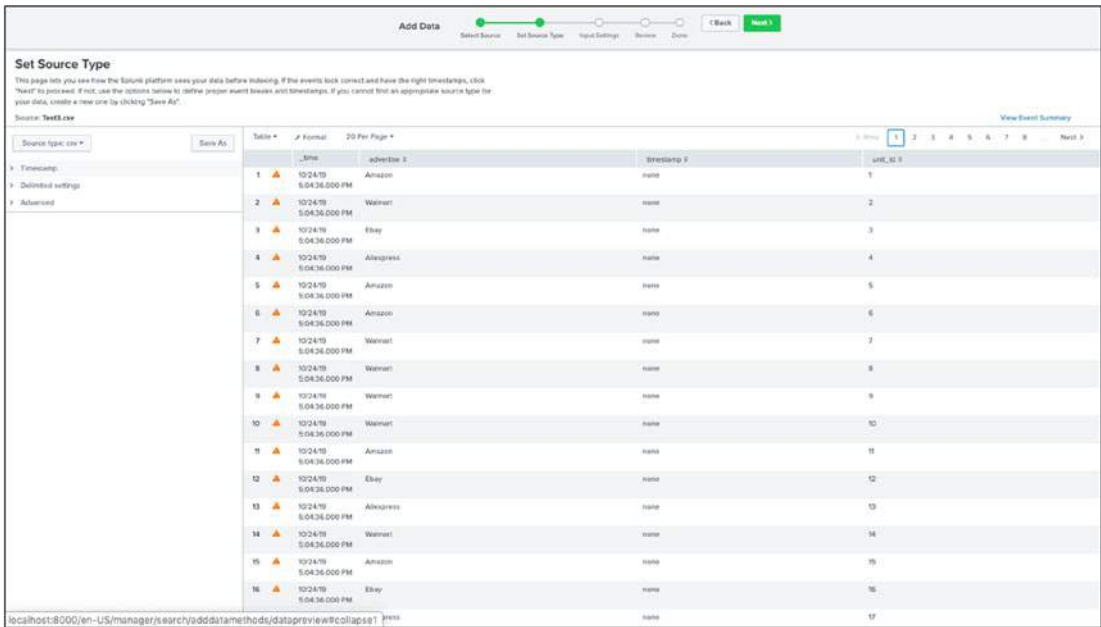


Figure 3-11. *Sourcetype:test2*

4. Click Next and create the new index as tes22, the app name as test.

For Test2 .txt field extraction, do the following.

1. Navigate to the test app search command. The index is tes22, and in the source is test2.
2. Use Delimiters Extract Field for Test2.txt in the Field name, and then provide the following field names and set Tab as the delimiter.

```
field1=unit_id
field2=method
field3=timestamp
field4=status
field5=category
field6=location
```

For Test3 .txt field extraction, do the following.

1. Navigate to the Test App using Splunk Web, traverse to search command and type there **index = "tes22" source = "Test3.csv"**.
2. Use the Delimiters Extract Field for Test3 in the Field name, and then provide the following field names and set the delimiter as Comma.

```
field1=id
field2=url
```

In Test2.txt, "unit\_id" is a common field, and in Test3.csv, "id" is a common field, so you can use field aliases to correlate events.

After entering the credentials, follow these steps.

1. Go to Settings.
2. Add Fields.
3. Go to Field aliases.
4. Add New.
5. In Destination app, type test.
6. In Name, type **tes11\_test**.
7. In Apply to, select source.
8. In named, type test3.csv.
9. In Field aliases, type **id=ptestid**.

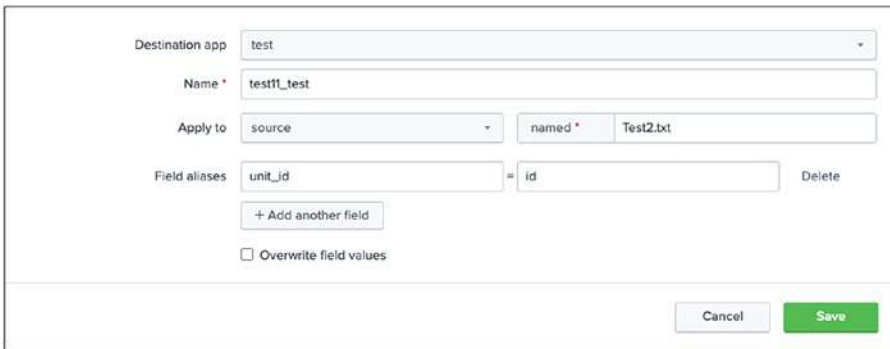
The Add New screen is shown in Figure 3-12.

**Figure 3-12.** Field aliases tes11\_test:test3.csv

Like the prior example, add another field alias having similar field values.

1. In Destination app, type test.
2. In Name, type **tes11\_test**.
3. In Apply to, select Source, and in named, type Test3.csv.
4. In Field aliases, type **unit\_id=ptestid**.

The working screen is shown in Figure 3-13.



**Figure 3-13.** Field aliases tes11\_test:Test3.csv

To check the success of the commands, refer to the following steps.

1. Go to Settings and then go to Fields.
2. On the Field Aliases page, you find that two field aliases have been added, as shown in Figure 3-14.

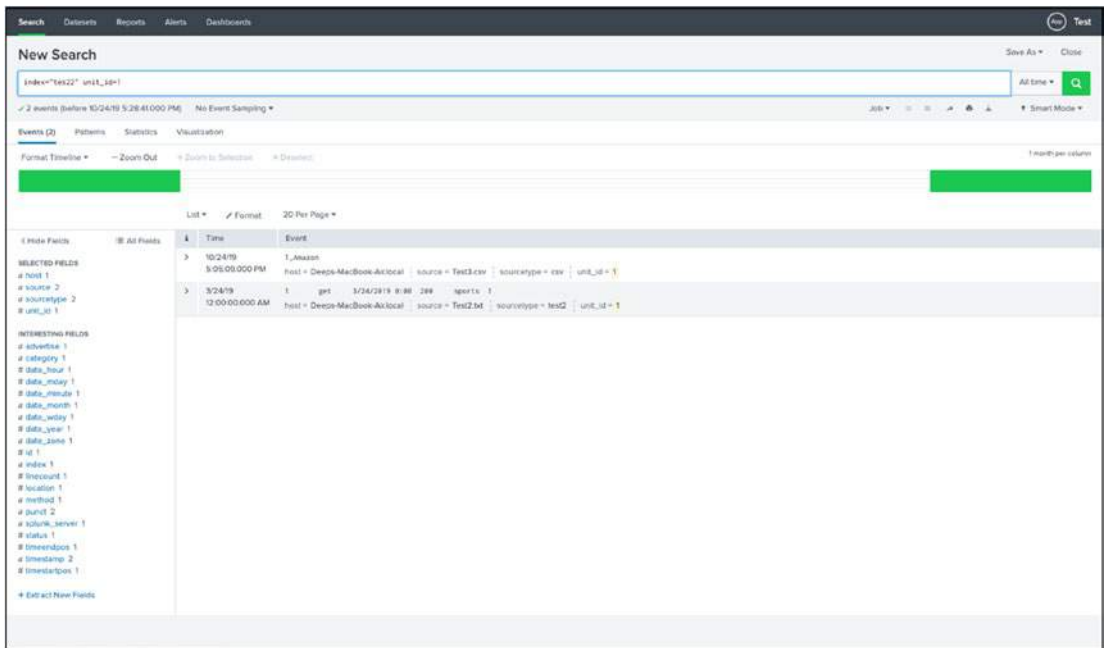
Name	Field aliases	Owner	App	Sharing	Status	Actions
Field Alias: test_app_access	test as test	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	made as file_id	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	vendor_object as file_name	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	vendor_object_path as file_path	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	file as file_size	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	tags as object_attrs	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	vendor_object as object	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	vendor_object_path as object_path	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	vendor_status as status	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	url as url_id	No owner	Splunk_SA_CIM	Global   Permissions	Enabled	Clone
Field Alias: test_app_syslog_change	unit_id ANNEW id	admin	test	Private   Permissions	Enabled	Clone   Merge   Delete

**Figure 3-14.** Illustrated Field Aliases: test app

To test the field aliases that you added, type the following SPL command in the search bar.

```
index="tes22" unit_id=1
```

The page to test the field aliases is shown in Figure 3-15.



**Figure 3-15.** Find Aliases `unit_id=1`

Field aliases provide normalization over any field (host, source, or source type)

Next, let's walk through a Splunk search query that helps correlate data.

## Splunk Search Query

Splunk search queries correlate data and are used in data searching, filtering, modification, manipulation, enrichment, insertion, and deletion. They provide flexibility when using small and big data. You can also analyze and visualize data in Splunk.

1. Write an `index="test89"` query in which the user wants to know at what time the total events count exceeds 150 within a 1-hour time span. Display the total count and the time when events exceed the limit.

This is the solution.

```
index="test89" | bin span=1h _time | stats count by _time | where count > 150
```

The solution is shown in Figure 3-16.



**Figure 3-16.**

2. Display transaction events on index="test89" and the maximum pause in a transaction is up to 10 minutes.

This is the solution.

```
index="test89" | transaction maxpause=10m
```

A solution screen is shown in Figure 3-17.



The screenshot shows the Splunk search interface. The search bar contains the query `index="test89" | transaction maxpause=10s`. Below the search bar, there are tabs for Events (15), Patterns, Statistics, and Visualization. The Events tab is active, showing a list of events. The table has columns for Time and Event. The Event column contains detailed log entries with fields like Data, Id, Client, and Inserted On.

Time	Event
1/30/19 6:39:27000 PM	Data Id Data Client Inserted On host = Deeps-MacBook-Airlocal source = Test (f.txt) sourcetype = Test5
12/27/18 11:00:00000 PM	1686 *SEIT_907409,40249,E,0,0,1,0,0,16784249,0,9,A,19.000027,72.849663,16,200,173069,271218,24,1,14,0,4203,1142,0,1,0,0,0,4.2,E10.21,0 27.57.85.135 - 55333 12/27/2018 23:00 1687 *SEIT_907409,40250,E,0,0,1,0,0,16784249,0,9,A,19.000027,72.849663,16,200,173109,271218,24,1,14,0,4203,1142,0,1,0,0,0,4.2,E10.21,0 27.57.85.135 - 55333 12/27/2018 23:01 1688 *SEIT_907409,40251,E,0,0,1,0,0,16784249,0,9,A,19.000027,72.849663,16,200,173209,271218,24,1,14,0,4214,1142,0,1,0,0,0,4.2,E10.21,0 27.57.85.135 - 55333 12/27/2018 23:02 1689 *SEIT_907409,40252,E,0,0,1,0,0,16784249,0,9,A,19.000027,72.849663,16,200,173309,271218,24,1,14,0,4205,1142,0,1,0,0,0,4.2,E10.21,0 27.57.85.135 - 55333 12/27/2018 23:03 1690 *SEIT_907409,40253,E,0,0,1,0,0,16784249,0,10,A,19.000027,72.849663,16,200,173409,271218,24,1,14,0,4200,1142,0,1,0,0,0,4.2,E10.21,0 27.57.85.135 - 55333 12/27/2018 23:04
12/27/18 10:00:00000 PM	1621 *SEIT_907409,40144,A,0,1,1,0,0,16784228,0,8,A,19.079047,72.849820,65525,337,163005,271218,31,1,14,0,4203,1153,0,1,0,0,0,4.2,E10.21,0 27.57.85.135 - 55333 12/27/2018 22:00 1622 *SEIT_907409,40145,A,0,1,1,0,0,16784228,0,9,A,19.079047,72.849820,65525,337,163105,271218,31,1,14,0,4205,746,0,1,0,0,0,4.2,E10.21,0 27.57.85.135 - 55333 12/27/2018 22:01 1623 *SEIT_907409,40146,S,0,1,1,0,0,16784231,10,9,A,19.079282,72.849867,39,358,163159,271218,29,1,14,0,4214,1241,0,1,0,0,0,4.2,E10.21,0 27.57.85.135 - 55333 12/27/2018 22:02 1624 *SEIT_907409,40147,A,0,1,1,0,0,16784231,0,9,A,19.079430,72.849872,41,9,163205,271218,29,1,14,0,4222,1249,0,1,0,0,0,4.2,E10.21,0 27.57.85.135 - 55333 12/27/2018 22:02

**Figure 3-17.**

- Write a Splunk query on `index="test89"` to get the event count from each port.

This is the solution.

```
index="test89" |stats count by port
```

The solution is shown in Figure 3-18.

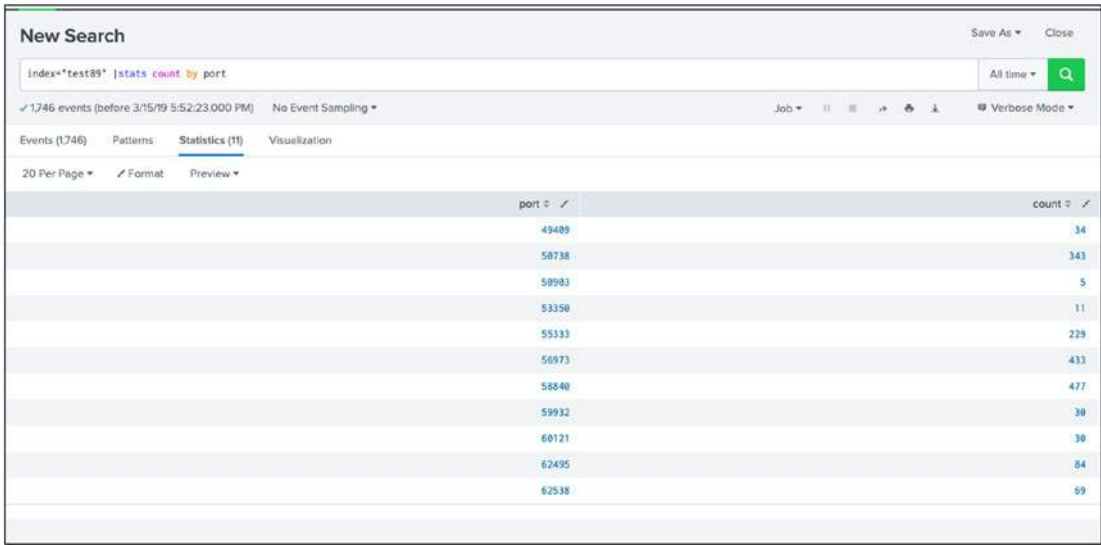


Figure 3-18.

- 4. Write a Splunk query on index="test89" to get a session where the timespan is 1 hour.

This is the solution.

```
index="test89" | `session` | timechart span=1h sum(eventcount) as sessions
```

The solution is shown in Figure 3-19.

The screenshot shows the Splunk Search interface for a new search. The search query is `index="test89" | session | timechart span=1h sum(eventcount) as sessions`. The results are displayed as a timechart with a grid. The x-axis represents time in 30-minute intervals from 2018-12-26 23:30:00 to 2018-12-27 12:30:00. The y-axis represents the number of sessions, with a scale from 0 to 60. The data shows a steady increase in sessions over time, starting at 0 and reaching 60 by the end of the period.

Time	Sessions
2018-12-26 23:30:00	0
2018-12-27 00:30:00	0
2018-12-27 01:30:00	0
2018-12-27 02:30:00	0
2018-12-27 03:30:00	0
2018-12-27 04:30:00	0
2018-12-27 05:30:00	0
2018-12-27 06:30:00	0
2018-12-27 07:30:00	0
2018-12-27 08:30:00	0
2018-12-27 09:30:00	0
2018-12-27 10:30:00	60
2018-12-27 11:30:00	60
2018-12-27 12:30:00	60

**Figure 3-19.**

Pat yourself on the back because you have successfully learned how to create macros and field aliases in Splunk using regular expressions and delimiters.

## Summary

This chapter covered field extraction in Splunk to extract data fields during indexing and searching. You learned that macros are useful when you need to frequently run the search command because it prevents rewriting the entire command. You also learned about field aliases, which correlate various events with similar field values. Also, a Splunk search query provides flexibility when working with small and big data.

According to the Splunk Power User exam blueprint, 4%–10% is on field extraction, 7%–10% is on macros, and 5%–10% is on field aliases.

In the next chapter, you learn about Splunk tags and lookups and create and alerts.

## Multiple Choice Test Questions

- A. If there is a conflict between a whitelist and a blacklist input setting, which one is used?
  - 1. blacklist
  - 2. whitelist
  - 3. they cancel each other out
  - 4. none of the above
  
- B. Which parent directory contains the configuration files in Splunk?
  - 1. SPLUNK\_HOME/etc
  - 2. SPLUNK\_HOME/var
  - 3. SPLUNK\_HOME/conf
  - 4. SPLUNK\_HOME/default
  
- C. Once a regular expression is edited, can you go back to the Field Extractor UI?
  - 1. True
  - 2. False
  
- D. Are delimiters mostly used in structured data?
  - 1. True
  - 2. False
  
- E. Field aliases normalize data over which default fields? (Select all that apply.)
  - 1. Host
  - 2. Source
  - 3. Source type
  - 4. Events

- F. Both fields appear in the All Fields and Interesting Fields lists, if they appear in at least \_\_\_\_\_ of events.
1. 10%
  2. 80%
  3. 40%
  4. 20%

### Answers

A: 1

B: 1

C: 1

D: 2

E: 1, 2, and 4

F: 4

## References

- *Splunk Operational Intelligence* by Josh Diakun (Packt Publishing, 2018)
- <https://docs.splunk.com/Documentation/UnixApp/5.2.5/User/Searchmacros>
- <https://docs.splunk.com/Documentation/Splunk/7.3.1/Knowledge/Definesearchmacros>
- <https://docs.splunk.com/Documentation/Splunk/7.3.1/Knowledge/Managesearch-timefieldextractions>

## CHAPTER 4

# Tags, Lookups, and Correlating Events

In the previous chapter, you learned how to extract fields from Splunk using delimiters and regular expressions to create macros and field aliases. You also performed a few Search Processing Language commands to improve retention rates. In this chapter, you deal with Splunk tags and lookups and create various reports and alerts. Splunk tags are pairs of nomenclature-added values that assign names to a specific field and its value combination. Splunk lookups enhance data by adding a field-value combination from any external data source. Reports are saved search results, which represent related statistics and visuals. Meanwhile, alerts are triggered when a search result satisfies any condition.

The following are the topics in this chapter.

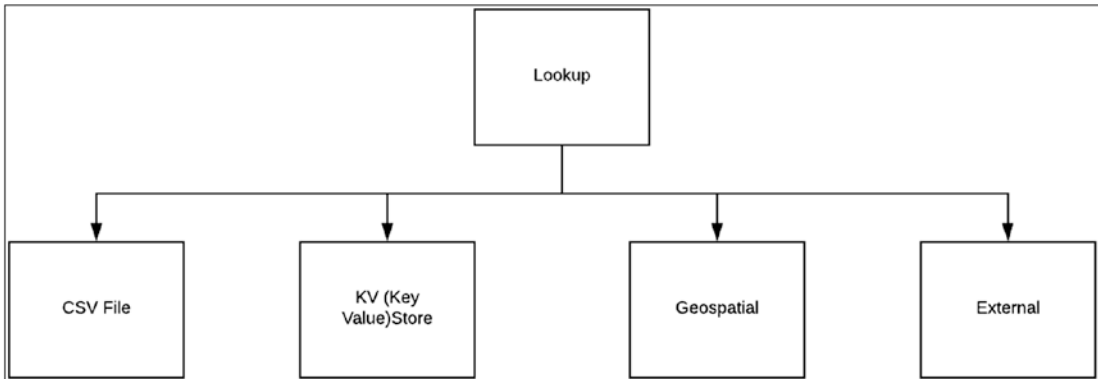
- Splunk lookups
- Splunk tags
- Creating reports in Splunk
- Creating alerts in Splunk

By the end of this chapter, you'll have covered at least 10% of the Splunk Power User exam blueprint and 21% of the Splunk User exam blueprint.

## Splunk Lookups

**Lookups** are the commands that enhance data through the **addition of field-values to existing data**. Data from multiple sources are added to an existing source to enhance it with lookup map values in an event with the relevant field in another data source. For example, you can use a lookup to match an error code from a browser and return a new

field with a detailed description of the error code. There are 4 types of lookups in Splunk, as represented in Figure 4-1.



**Figure 4-1.** *Splunk lookups*

Each of these lookup types is explained as follows.

- **CSV lookups** pull key-value pairs from CSV files. They add data from events with fields and represent a static table of data. Therefore, they are also called **static lookups**. Each column in a CSV table is interpreted as a potential field value.
- **KV Store lookups** pull key-value pairs from a defined app and add the event data with fields pulled from your app’s Key-Value Store (KV Store) collections.
- **Geospatial lookups** create queries that return results, which Splunk software can use to generate a choropleth map visualization. In this type of lookup, the data source is a **KMZ** (compressed keyhole markup language) file, which defines mapped regions’ boundaries.
- **External lookups** populate event data from an external source. They can be **Python scripts** or **binary executables** that get field values from an external source. They are also called **scripted lookups**.

There are three different ways to create a lookup in Splunk.

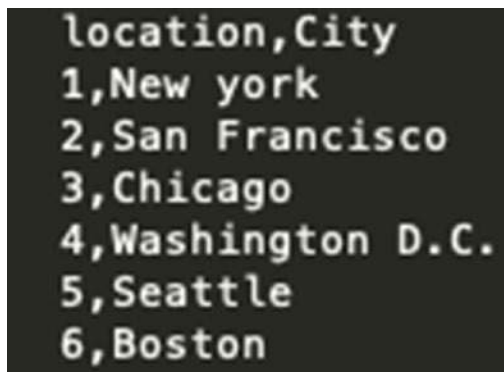
- Lookup table files
- Lookup definitions
- Automatic lookups

Field values are case sensitive by default.

## Looking up Table Files

You use the lookup table file in Splunk when you need to define a lookup using a CSV file or a GZ file, depending on their availability. In this section, you define a sample lookup table, which is called `Location.csv`. You can define your files in a preexisting lookup table file.

1. Download `Location.csv` from <https://github.com/deeppmehta/splunk-certification-guide/blob/main/ch4/Location.csv>. `Location.csv` has two columns—location and state, as shown in Figure 4-2.



```
location, City
1, New york
2, San Francisco
3, Chicago
4, Washington D.C.
5, Seattle
6, Boston
```

*Figure 4-2. Lookup table*

2. Go to Settings on Splunk Web, and then go to Lookups.
3. Go to Lookup Table Files and click Add New.
4. In Destination app, select the test app.



5. Upload the Location.csv file. For your reference, a sample image is shown in Figure 4-3.

The screenshot shows a web form for creating a lookup table in Splunk. It has three main sections: 1. 'Destination app' with a dropdown menu showing 'test'. 2. 'Upload a lookup file' with a 'Choose file' button and the text 'Location.csv'. Below this is a note: 'Select either a plaintext CSV file, a gzipped CSV file, or a KMZ/KML file. The maximum file size that can be uploaded through the browser is 500MB.' 3. 'Destination filename \*' with a text input field containing 'Location.csv'. Below this is a note: 'Enter the name this lookup table file will have on the Splunk server. If you are uploading a gzipped CSV file, enter a filename ending in ".gz". If you are uploading a plaintext CSV file, we recommend a filename ending in ".csv". For a KMZ/KML file, we recommend a filename ending in ".kmz"/".kml".' At the bottom right, there are two buttons: 'Cancel' and 'Save'.

**Figure 4-3.** Splunk lookup table file

6. Once the lookup is created, go to Lookup Table Filter, where you find Location.csv.
7. Click Permission on the Location.csv file. You have the option to edit various roles. Select the **Read and write to everyone** option. (Chapter 6 covers knowledge object permissions.)

CSV files and GZ files are only used for lookup tables.

## Lookup Definitions

Lookup definitions define the data types and connection parameters that compare event fields in Splunk. Lookup definitions require a lookup table. In this section, you create a lookup named *countries* for the Location.csv lookup file defined in the lookup tables file.

Follow these steps to create lookup definitions.

1. Go to Settings and select Lookups.
2. Go to Lookup Definitions and click Add New.
3. In Destination app, select the test app. Name it **Countries**. In Type, select File-based. In Lookup file, select Location.csv. For a better understanding, refer to Figure 4-4.

Destination app: test

Name \*: Countries

Type: File-based

Lookup file \*: Location.csv

[Create and manage lookup table files.](#)

Configure time-based lookup

Advanced options

Cancel Save

**Figure 4-4.** Lookup definitions

4. Click the Save button and edit the countries file's permissions to allow read and write to everyone.

In the next section, you map the lookup input field with the lookup output field.

## Automatic Lookups

In an automatic lookup, the table file requires the lookup input field to be mapped with the lookup output field. In this section, you depict the field location events from the Test2.txt source type with the States column in the Location.csv.

The following steps create an automatic lookup.

1. Go to Settings and click Lookups.
2. Select Automatic Lookups and then select Add New.
3. In Destination app, select test. In Name, enter **test**. In Lookup table, select the countries that you created in the lookup definitions. In Apply to, select source type named as Test2.txt. In Lookup input fields, type **location = location**. In Lookup output fields, enter **City = City**. Refer to Figure 4-5.

The screenshot shows the configuration interface for automatic lookups in Splunk. It includes the following fields and options:

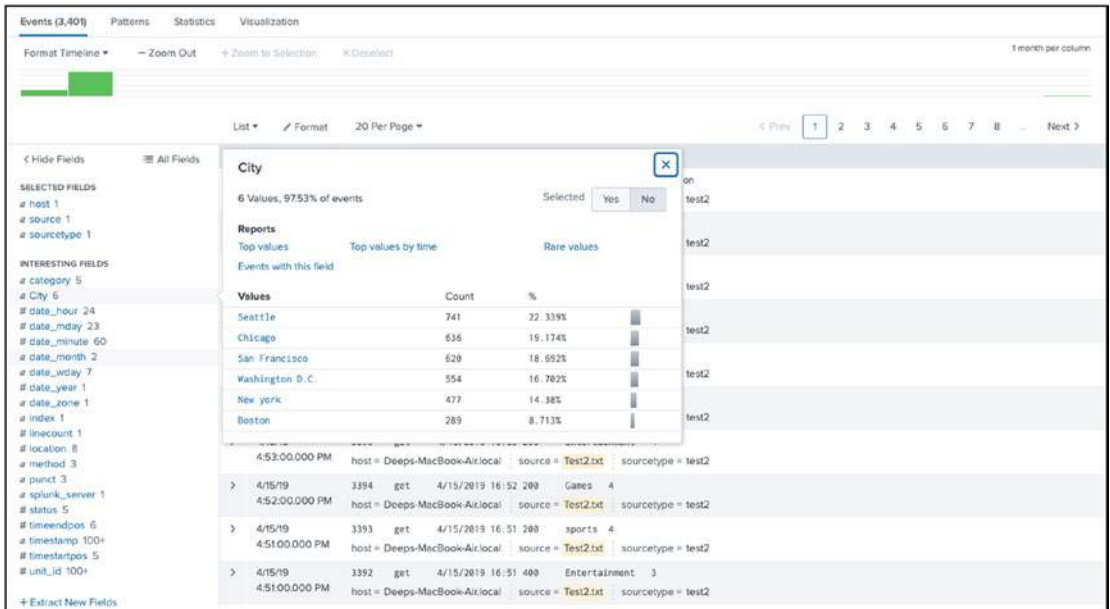
- Destination app:** A dropdown menu set to "test".
- Name:** A text input field containing "test".
- Lookup table:** A dropdown menu set to "Countries".
- Apply to:** A dropdown menu set to "source" and a "named" checkbox checked, with a text input field containing "Test2.txt".
- Lookup input fields:** A table with one row: "location" in the input field, "=" in the middle, "location" in the output field, and "Delete" on the right. Below this is a "+ Add another field" button.
- Lookup output fields:** A table with one row: "City" in the input field, "=" in the middle, "City" in the output field, and "Delete" on the right. Below this is a "+ Add another field" button.
- Overwrite field values:** An unchecked checkbox.
- Buttons:** "Cancel" and "Save" buttons at the bottom right.

**Figure 4-5.** Splunk automatic lookups

4. In the test app search bar, type the following command.

```
index="tes22" source="Test2.txt"|lookup Location.csv  
location output City
```

You can find states in a field with events mapped (see Figure 4-6).



**Figure 4-6.** Splunk lookup

In Splunk, admins can change the `case_sensitive_match` option to `false` in `transforms.conf`.

## Splunk Tags

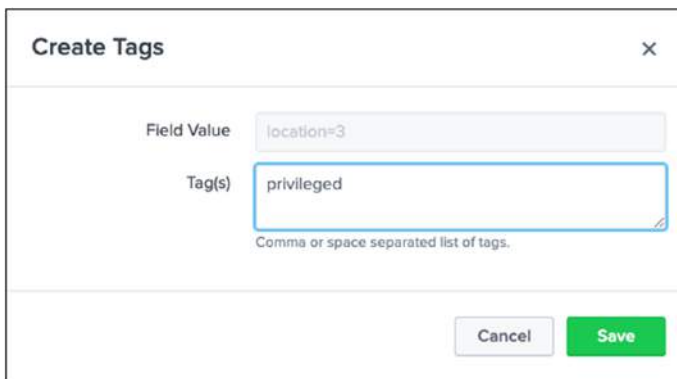
Tags are also knowledge objects in Splunk. They allow the **user to search for an object with a particular field value**. Tags make data more understandable. More than one tag can be created for a specific field-value combination. Tags are also responsible for assigning case-sensitive names to field-values.

## Create Tags in Splunk Using Splunk Web

The Splunk Web interface conveniently creates tags. In this section, you create a tag named *privileged* for location=3 in the Test2.txt source. Follow these steps to create tags.

1. Go to the test app. In the search bar, enter **Type** followed by this command:  

```
index="Test" source="Test2.txt"
```
2. Explore the event and go to field named Location.
3. In Actions, click the down arrow and select Edit Tags.
4. In Tag(s) type **privileged**. The Field Value is location=3, as shown in Figure 4-7.



**Figure 4-7.** *Splunk tags*

5. You have successfully created a tag. You can check it with the following search command.

```
(index="test" source="Test2.txt") "tag::location"=privileged
```

---

You can use \* in wildcards and for tags associated with field value  
tag::tag\_name=tag\_value.

---

## Tag Event Types in Splunk Web

Tag event types in Splunk add extra information to events. In this section, tag event type named *privileged* is located in the Test2.txt source.

Follow these steps to create this.

1. Go to Settings and then select Event types.
2. Create a new event type.
3. In Destination App, select test. In Name, enter **test0102**. In Tag(s), enter **privileged**. In Color, select red. In Priority, select 1. In Search string, type the search query as follows.

```
(index="test" source="Test2.txt") "tag::location"=privileged
```

Figure 4-8 represents the working page.

The screenshot shows the 'Create New Event Type' form in Splunk Web. The fields are filled as follows:

- Destination App: test
- Name: test0102
- Search string: (index="test" source="Test2.txt") "tag::location"=privileged
- Tag(s): privileged
- Color: red
- Priority: 1 (Highest)

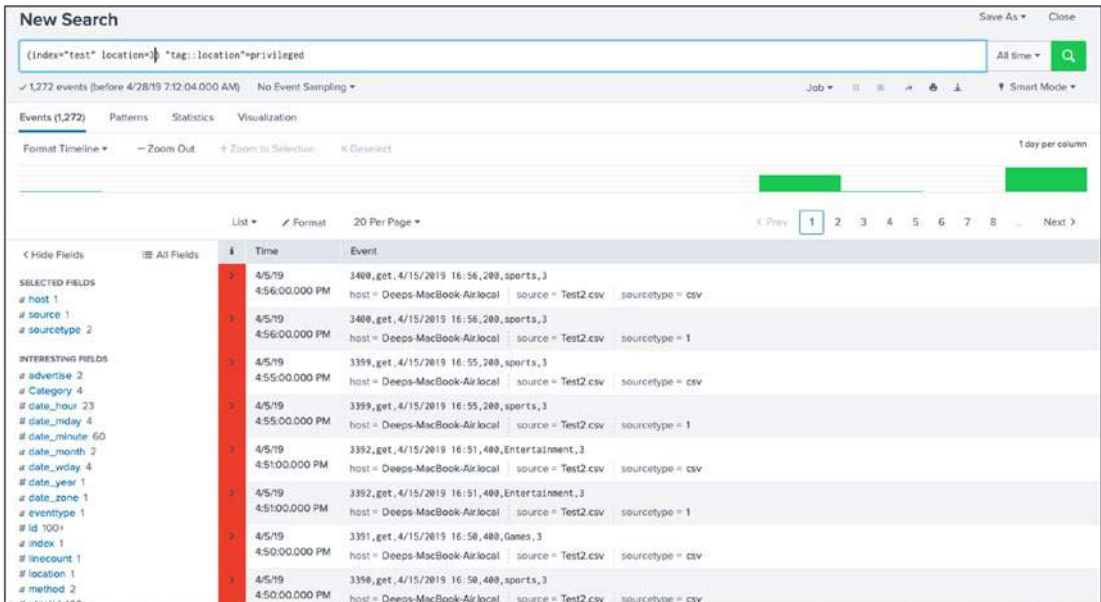
Buttons for 'Cancel' and 'Save' are visible at the bottom right of the form.

**Figure 4-8.** Splunk tag using event type

4. Click Save.
5. If you call all the events with a privileged tag in the search bar, you find all events that are priority 1 and the color red.

`(index="test" location=3) "tag::location"=privileged`

All the events that are priority 1 and the color red, as stated in the search query, are shown in Figure 4-9.



**Figure 4-9.** Splunk tag using event type

Set the priority and color based on the specific event’s importance. If you find a particular event that is more important, give it a higher priority (the opposite for a low priority event).

The following section covers the process of reporting in Splunk.

# Reporting in Splunk

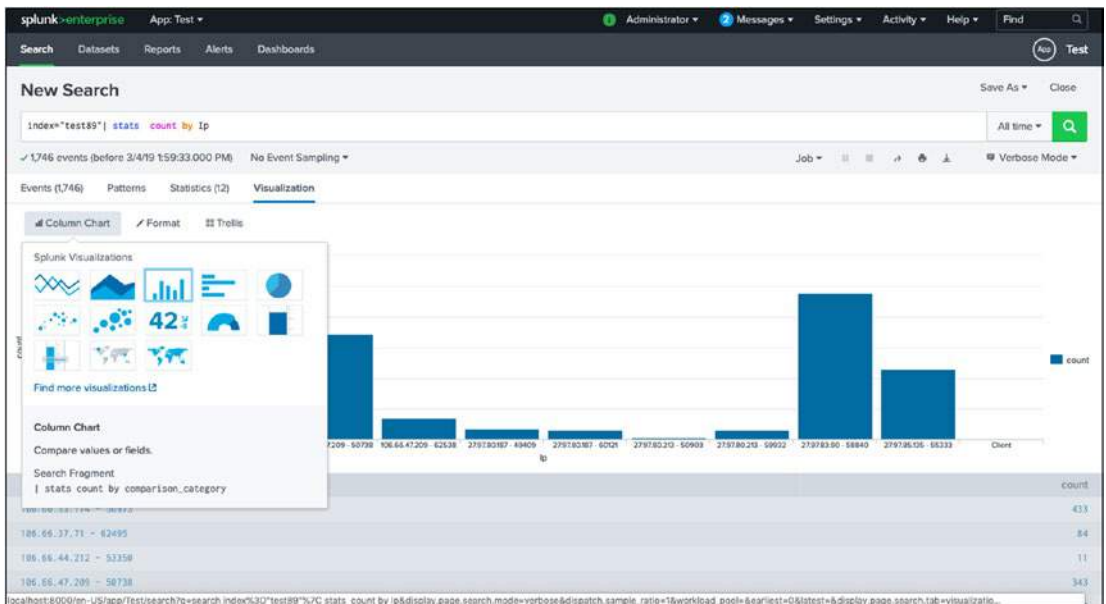
Reports are results that contain defined information represented through tables, charts, and so forth. They are created through a saved search command for reuse. Reports in Splunk can be configured to perform versatile operations and can be converted into PDF format and sent as email. Reports can also be scheduled in Splunk.

Let's start with learning how to create reports in Splunk Web.

## Creating Reports in Splunk Web

Reports reuse Splunk search commands (Splunk Search Processing Language). In this section, you create a report on the number of received events from `index=test89`.

1. Navigate to the test app in Splunk.
2. Type the following search command.  
`index=test89| stats count by Ip`
3. In Time Range Picker, select All Time.
4. Go to Visualization and select Column Chart. The output is shown in Figure 4-10.



**Figure 4-10.** Splunk visualization



5. In the upper right corner, click Save As.
6. In Title, type **Test1**. In Content, select table + chart (see Figure 4-11).

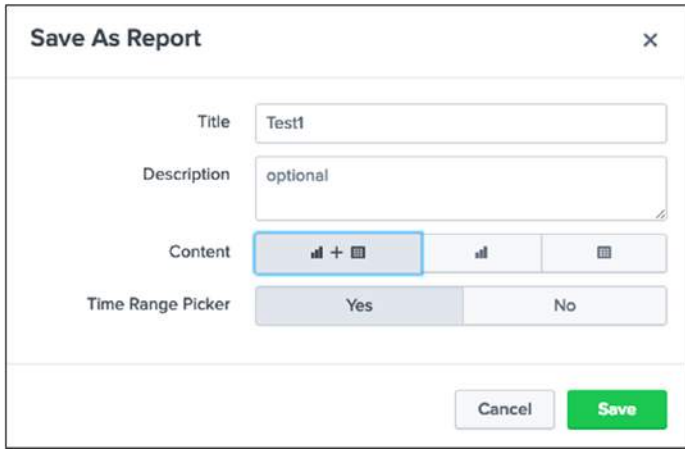


Figure 4-11. Splunk report

The Test1 report is shown in Figure 4-12.

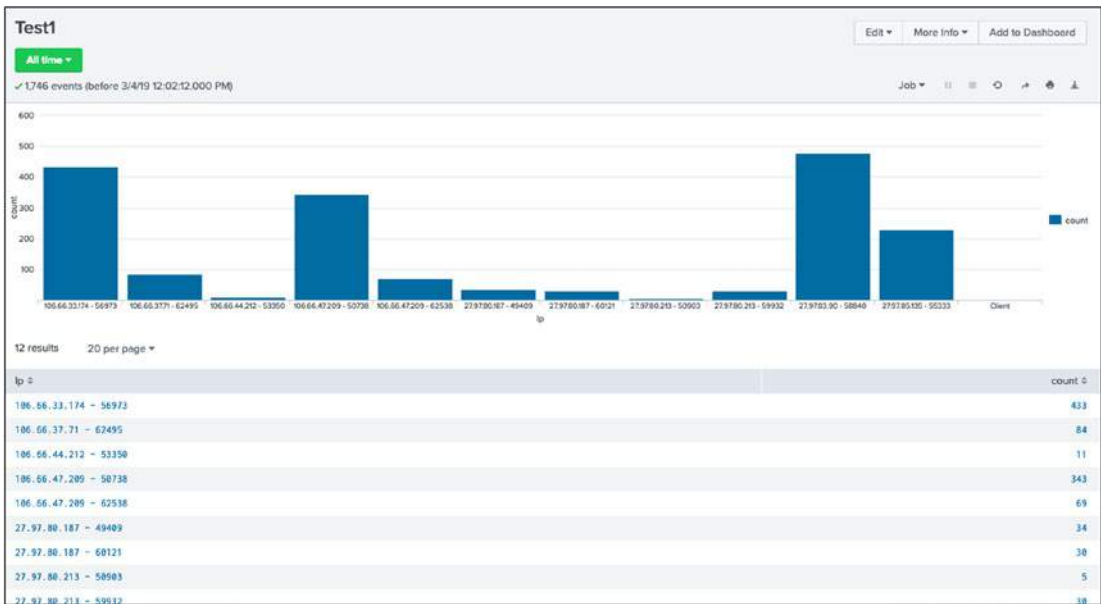


Figure 4-12. Splunk Test1 report

You can also create a report by going to Settings ► Searches ► Reports and Alerts ► Create a New Report. In this report, you may also get two extra fields: the earliest time and the latest time. Figure 4-13 represents the working page of a new report named *test* where the earliest time is *-1y*.

The screenshot shows the 'Create Report' dialog box in Splunk. The dialog is titled 'Create Report' and has a close button (X) in the top right corner. It contains the following fields and options:

- Title:** A text input field containing the value 'test'.
- Description:** A text input field containing the value 'optional'.
- Search:** A text input field containing the search query 'index="test89" | stats count by Ip'.
- Earliest time:** A text input field containing the value '-1y'. This field is highlighted with a blue border. Below it, there is a link for 'Time specifiers: y, mon, d, h, m, s Learn More'.
- Latest time:** A text input field containing the value 'optional'. Below it, there is a link for 'Time specifiers: y, mon, d, h, m, s Learn More'.
- App:** A dropdown menu currently showing 'Test (Test)'.
- Time Range Picker:** A button group with two options: 'Yes' (which is selected) and 'No'.

At the bottom of the dialog, there are two buttons: 'Cancel' and 'Save'.

**Figure 4-13.** Report in Splunk

Next, you learn about report acceleration in Splunk.

## Report Acceleration in Splunk

Report acceleration in Splunk means systematically running reports with large datasets covering long data ranges. When you accelerate a report, a summary is created; it comes in handy when processing large volumes of data. The acceleration search mode must be either smart or fast.

### Creating Report Acceleration

Report acceleration in Splunk creates a summary of the report. In this section, you accelerate the report to discover the ports where maximum events are incoming from `index="test89"`.

Follow these steps (also see Figure 4-14).

1. Create a report to learn the top ports for the index = test89. Name the report **test10**. In in-app, select the test app. Refer to the following query.

```
index="test89" | top port
```

2. Go to Settings. Select Searches, Reports, and Alerts.
3. In App, select test.
4. Go to the test10 report. Click Edit and go to Edit Acceleration.
5. Click Accelerate report.
6. Click Next.
7. In Summary Range, select All Time.

**Edit Acceleration** ×

Report **test10**

Accelerate Report

Acceleration might increase storage and processing costs. Acceleration can return invalid results if you change definitions of knowledge objects used in the search string after you accelerate the report. [Learn More](#)

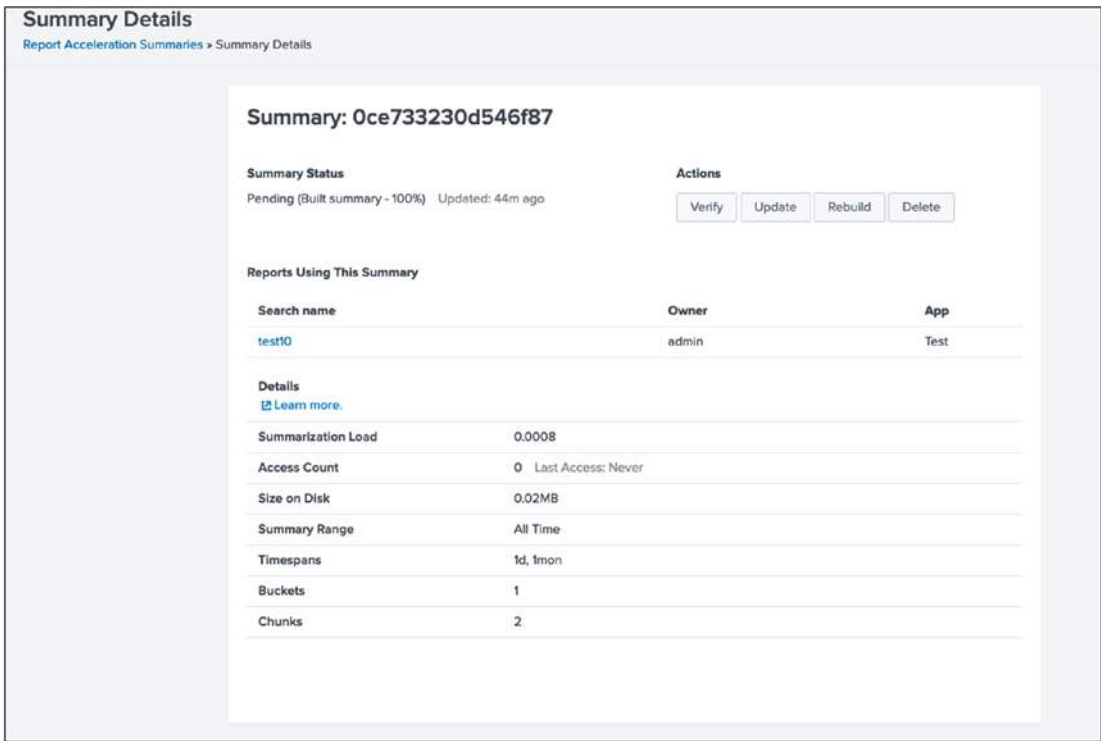
Summary Range ? All Time ▾

Cancel Save

**Figure 4-14.** Acceleration page

Do the following to check the report summary.

1. Go to Settings.
2. Select Report Acceleration and summary.
3. When you click Summary Id, you get content like event size on disk, summary range, buckets, chunks, and so forth (see Figure 4-15).



**Figure 4-15.** Sample summary

Report acceleration in Splunk uses automatically created summaries to speed the process, but there is cost associated with it because the number of accelerated reports account to concurrent searches and could lead to increase usage in the indexer.

## Scheduling a Report in Splunk

A scheduled report is a predefined report that runs in fixed intervals. Once a report is triggered, you can define its actions into various stems, like sending a report summary by email, reporting the results through a CSV lookup file, using webhooks, or logging and indexing searchable events. In this section, you schedule a Test1 report and run it every Monday at 10:00 with the highest priority.

To do this, refer to the following steps.

1. Go to Settings. Select Searches, Reports, and Alerts.
2. In App, select test. Go to test1 report. Click Edit and select Edit Schedule.
3. In Edit Schedule, select Run every week on Monday at 10:00 and set priority to Highest (see Figure 4-16).

**Edit Schedule** ×

**⚠** Scheduling this report results in removal of the time picker from the report display.

Report: **Test1**

Schedule Report:  [Learn More](#)

Schedule: Run every week ▼

On: Monday ▼ at 10:00 ▼

Time Range: All time ▶

Schedule Priority ? Highest ▼

Schedule Window ? 5 minutes ▼

**Trigger Actions**

+ Add Actions ▼

Cancel Save

**Figure 4-16.** Schedule page

To edit a scheduled report, you need to get write access from the Splunk admin.

That completes the process of creating and reporting in Splunk Web. The following section discusses alerts.

## Alerts in Splunk

Alerts are very important in organizations, as they **provide information regarding the state of the system**. For instance, an alert can determine if the load on a server has increased or it may provide operational insights. In Splunk, alerts are saved searches that look for events in real time. They trigger when the search results meet a certain condition.

Let's start with learning how to create alerts in Splunk Web.

### Create Alerts in Splunk Using Splunk Web

Alerts in Splunk monitor and alert when search results meet certain conditions. Once the alert is triggered, you can define the actions. In this section, you create an alert that runs on the first day of the month at 0:00. You want to know when the top port's event count exceeds or is equal to 500.

Follow these steps to perform this task.

1. Go to Settings. Select Searches, Reports, and Alerts.
2. Select New Alerts.
3. Set the title as test1298 and the app name as test. Select run every month on day 1 at 0:00. Set port >=500. Refer to Figure 4-17.

**Create Alert** [X]

**Settings**

Title: test1298

Description: Optional

Search: index="test89" | top port

App: Test (Test) ▾

Permissions: Private | Shared in App

Alert type: Scheduled | Real-time

Run every month ▾

On day: 1 ▾ at: 0:00 ▾

**Trigger Conditions**

Trigger alert when: Custom ▾

port>500  
e.g. \*search count > 10\*. Evaluated against the results of the base search.

Cancel Save

**Figure 4-17.** Alert in Splunk

4. Create a trigger in Splunk, as shown in Figure 4-18. Set the severity as low.

**Trigger Actions**

+ Add Actions ▾

When triggered ▾

🔔 Add to Triggered Alerts Remove

Severity Low ▾

**Figure 4-18.** Trigger in alert



By default, everyone has access to read the alert. Power users and admins have explicit rights to write alerts in Splunk.

## Cron Expressions for Alerts

You can use cron expressions for alert scheduling in Splunk. In this section, you create a cron expression that has the alert run every 12 hours. The alert lets you know when the event count for the top port exceeds or is equal to 500.

The syntax for the cron expression is as follows.

- Minutes: 0-59
- Hours: 0-23
- Day of the month: 1-31
- Month: 1-12
- Day of the week: 0-6 (where 0 = Sunday)

The following explains how to edit this alert.

1. In the test app, go to Settings.
2. Select Searches, Reports, and Alerts.
3. Select Alerts.
4. Edit the alert name to test1298.
5. In the **Run every month** drop-down menu, select **Run on Cron Schedule**, as shown in Figure 4-19.

**Edit Alert**
✕

---

**Settings**

Alert **test1298**

Description

Search

Alert type

Scheduled

Real-time

Run every month ▾

On day  at

**Trigger Conditions**

Trigger alert when

e.g. "search count > 10". Evaluated against the results of the base search.

Trigger

Once

For each result

Throttle ?

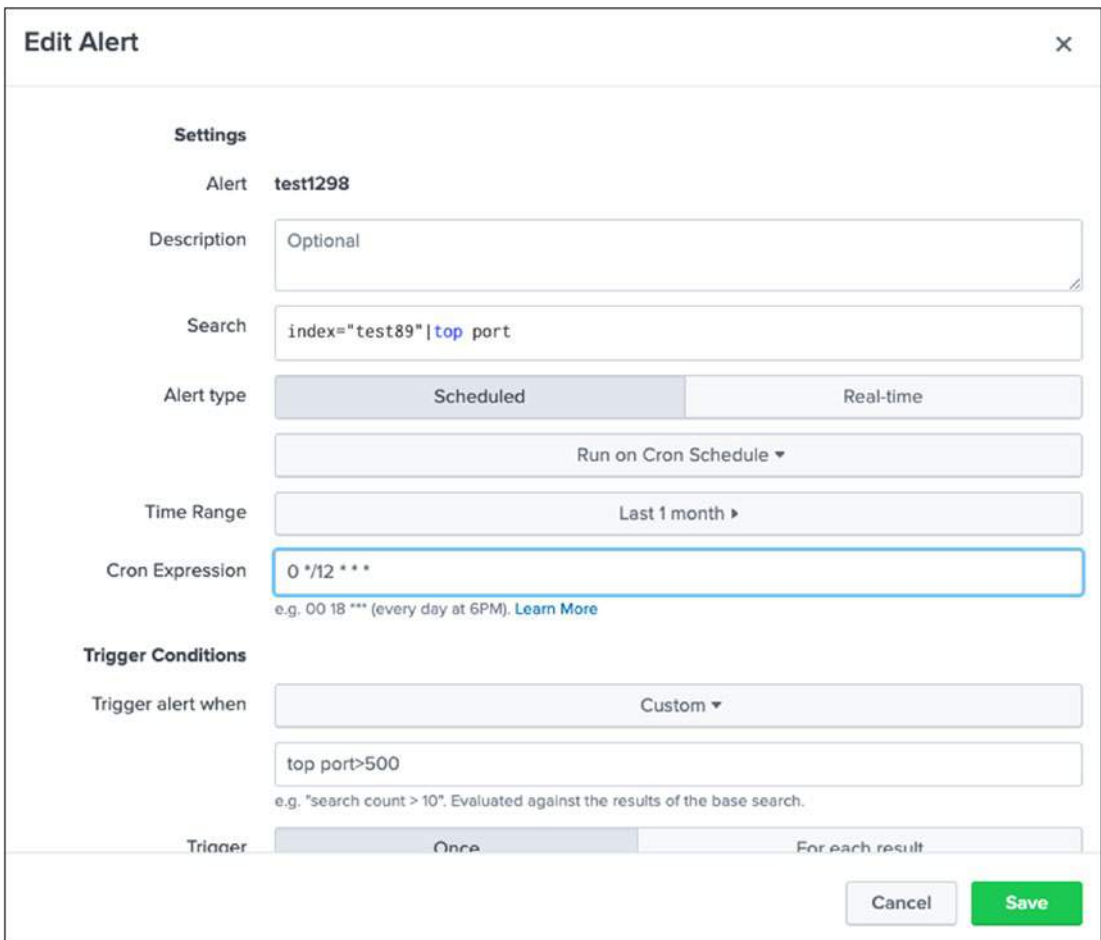
Cancel

Save

**Figure 4-19.** Alert page

You have created an alert that runs every 12 hours, as shown in Figure 4-20.

0 \*/12 \* \* \*



**Figure 4-20.** Cron expression

- 6. Click Save.

I would like to congratulate you for successfully learning about table files, definitions, and automatic lookups, and Splunk tags, reports, and alerts. Test your understanding with the MCQs at the end of the chapter.

## Summary

You studied lookups and their various types, like CSV lookups, external lookups, geospatial lookups, and KV Store lookups. You learned how to create tags in Splunk and how to generate reports and alerts in Splunk. These knowledge objects play a vital role, especially for the Splunk admin managing and processing large amounts of data.

Tags in Splunk, and according to Splunk Power User exam blueprint **Module 6-10%** for creating tags in Splunk. Lookups in Splunk, you also can create your lookups in Splunk and according to the Splunk User exam blueprint **Module 7-6%** for creating lookups in Splunk. Reports and alerts, you are also able to create your report and alerts in Splunk and according to Splunk User exam blueprint **Module 6,8-15%** for scheduling reports and alerts.

In the next chapter, you learn about data models, Pivot, event actions, and CIM.

## Multiple-Choice Questions

- A. Select all options that are lookup types.
  - 1. CSV lookup
  - 2. KV Store lookup
  - 3. external lookup
  - 4. None of the above
- B. In Splunk, field values are case insensitive by default
  - 1. true
  - 2. false
- C. In Splunk, tags use event types. You can set the priority and color based on an event's importance.
  - 1. true
  - 2. false
- D. If I am \_\_\_\_, I can edit a scheduled report.
  - 1. a user
  - 2. a power user
  - 3. an admin
  - 4. all of the above

- E. To create an alert, you need to be\_\_\_(Select all option that applies)
1. a user
  2. a power user
  3. an admin
  4. edit\_user
- F. To create an alert in Splunk using .conf edit \_\_\_\_.
1. savedsearches.conf
  2. props.conf
  3. alert.conf
  4. transformation.conf
- G. You can access a tag in Splunk using \_\_\_\_\_. (Select all that apply.)
1. .:
  2. =
  3. \*
  4. &

**Answers**

- a. 1, 2, 3
- b. 2
- c. 2
- d. 3
- e. 2, 3
- f. 1
- g. 1, 2, 3

## References

- <https://docs.splunk.com/Documentation/Splunk/7.2.4/Alert/CronExpressions>
- <https://www.youtube.com/watch?v=1IYezUcNGPY&t=1008s>
- *Splunk Operational Intelligence* by Josh Diakun (Packt Publishing, 2018)
- <https://docs.splunk.com/Documentation/Splunk/7.2.4/Knowledge/Abouttagsandalises>
- <https://docs.splunk.com/Documentation/Splunk/7.2.4/Report/Createandeditreports#>
- <https://docs.splunk.com/Splexicon:CommonInformationModel>

## CHAPTER 5

# Data Models, Pivot, and CIM

In the previous chapter, you learned about lookups and their various types. We also discussed tags, reports, and alerts in Splunk. In this chapter, you learn how to create a data model, an event action, and a Common Information Model.

Splunk software is artistic in its design, and it has a user-friendly interface that makes it easy to decipher commands and codes. The mark of a good software application is not whether it is easily understood by the computer but by the humans using it. Splunk offers adaptability through its functions, features, and commands. It is an easy platform to use with its vast scope and utility.

The following topics are covered in this chapter.

- Data models and Pivot
- Event action in Splunk
- Common Information Model in Splunk

A data model is a hierarchical normalized dataset. An event action is a highly configured knowledge object that communicates among the fields in Splunk Web and other configurable Internet sources. A CIM can normalize data using field names and event tags to get events from different data sources. By the time you complete this chapter, you will have embraced 30% of the Splunk Power User exam blueprint.

## Understanding Data Models and Pivot

Data models are **hierarchical datasets that enable Pivot users to create complex reports and dashboards** without writing complex commands in Splunk's Search Processing Language. When generating a report, data models and their respective datasets are encoded by the organization's knowledgeable managers.

The data model consists of events, transactions, or searches as a dataset.

Let's discuss the datasets and data models to better understand the SPL.

### Datasets and Data Models

Data models are composed of parent datasets and child datasets that are arranged sequentially. Each child dataset inherits constraints and fields from its parent dataset, besides having its own. (a concept similar to inheritance in programming).

A dataset is an **assemblage of data** that is defined and maintained for a specified purpose. It is depicted in a table comprising of fields in columns and their values in cells to be used in visually rich Pivot reports and to extend themselves during the search.

Datasets include auto-extracted fields, eval expressions, lookups, regular expressions, and geo IP fields.

### Creating Data Models and Pivot in Splunk

This section shows how data models are created and how Pivot recognizes a pattern of product sales requests from regions in the Eastern and Western United States.

#### Creating New Datasets

Follow these steps to create a new dataset.

1. Go to Settings and select the data models. Click Create a New Data Model. In Title, type **Country**. In App, select test (see Figure 5-1).



**New Data Model** [X]

Title

ID ?   
The data model ID can only contain letters, numbers, dashes, and underscores. Do not start the data model ID with a period.

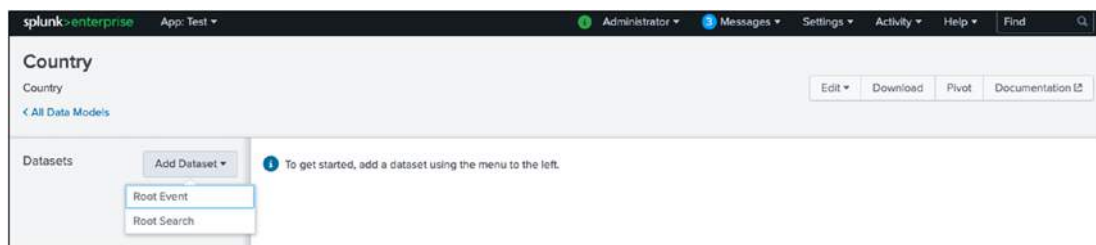
App

Description

**Figure 5-1.** Create Data Model:Country

2. Click Create. You are redirected to the dataset page.
3. Click Add Dataset and select Root Event from the drop-down menu.

The data model's Country page is shown in Figure 5-2.



**Figure 5-2.** Country Data Model

4. In the Dataset Name, enter **Country**, and in the Constraints, type the following search command.

```
index="tes22" source="Test2.txt" location=*
```

Figure 5-3 shows when the Country dataset is added.

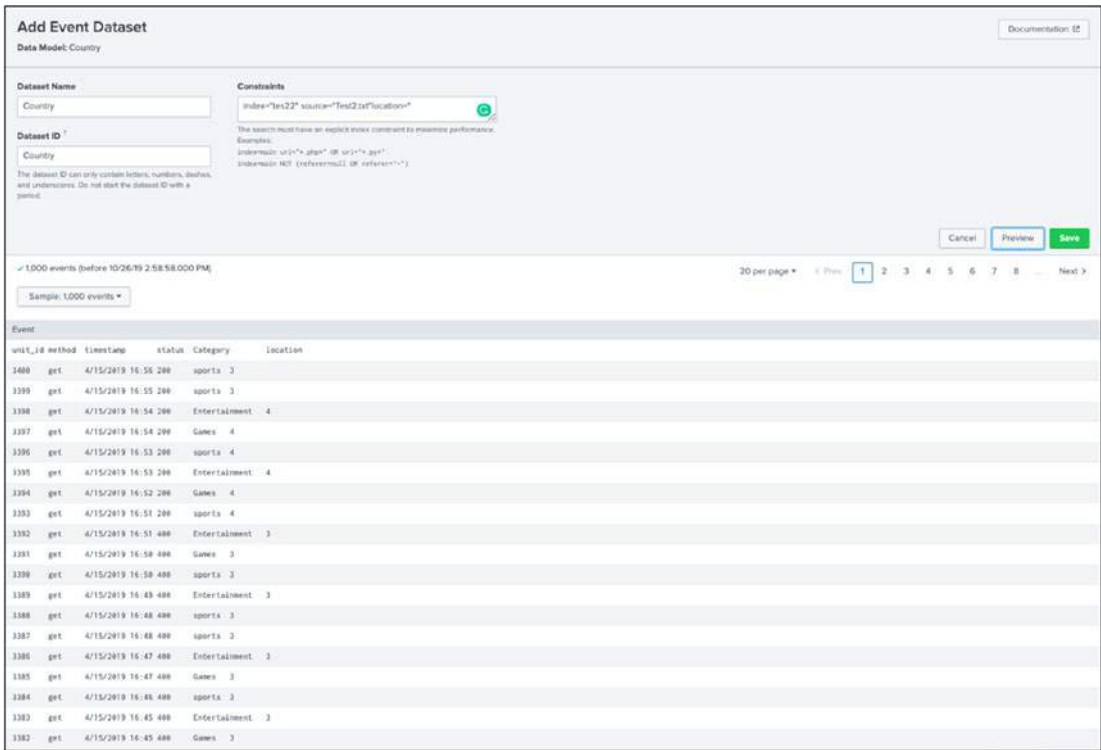


Figure 5-3. Data Model Country:Dataset

5. Save the settings.

Constraints in the dataset are essentially searched terms.

6. Select Add Fields Auto-Extracted.
7. Select Location From Auto-Extracted field.
8. Save the settings.

You have created a dataset. Let’s now move on to predicting sales patterns in the US regions.

## Predicting a Sales Pattern

To predict sales patterns in the Eastern and Western United States, you need to specify the states from Lookup - Country.

Follow these steps.

1. Go to Add Field and select the lookup. If Lookup - Country is not found, go to Settings ► Lookups ► Lookup table file ► New and define the lookup in that section.
2. In the Lookup table, select Countries.
3. Confirm that Field in Lookup is location.
4. Change Field in Dataset to location.
5. Select the City output field.
6. In Display Name, enter City.

These are shown in Figure 5-4.

**Lookup Table**

Countries ▼

**Input**

Field in Lookup: location ▼ = Field in Dataset: location ▼ Remove

Add New

**Output**

Field in Lookup:	Field in Dataset:	Display Name:	Type:	Flags:
<input type="checkbox"/> location	location		String ▼	Optional ▼
<input checked="" type="checkbox"/> City	City	City	String ▼	Optional ▼

Cancel Preview Save

**Figure 5-4.** Lookup Field:City

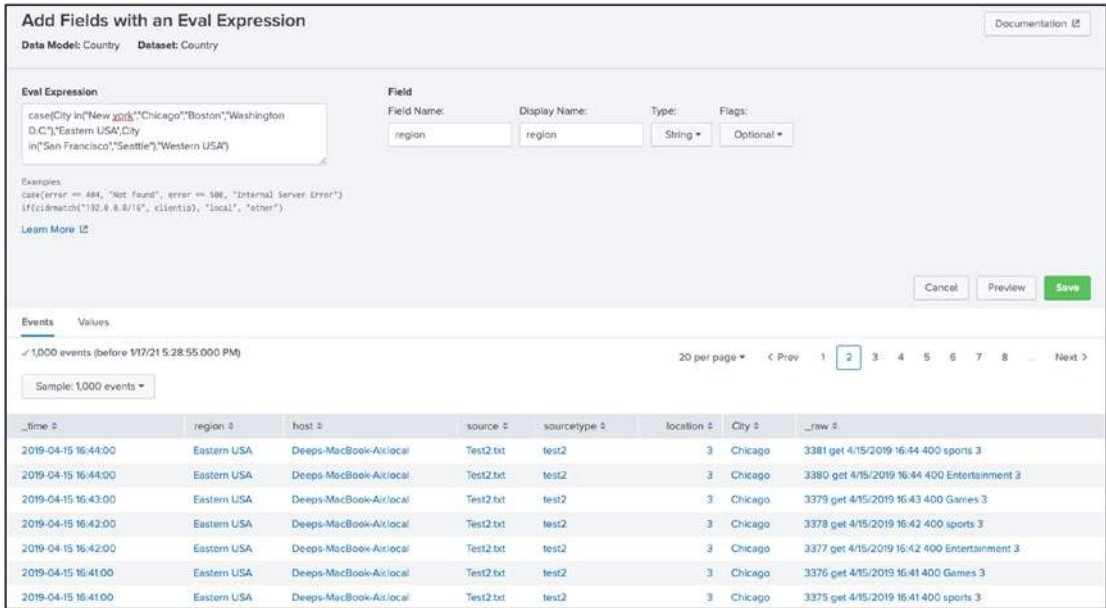
7. Save the changes.

Now, you need to appropriately categorize the states into Eastern USA or Western USA. For this, you are using evaluate expressions.

1. Click Add Field and Evaluate Expression.
2. In Field Name enter **region** and in Display Name enter **region**. In Eval Expression, type the following command.

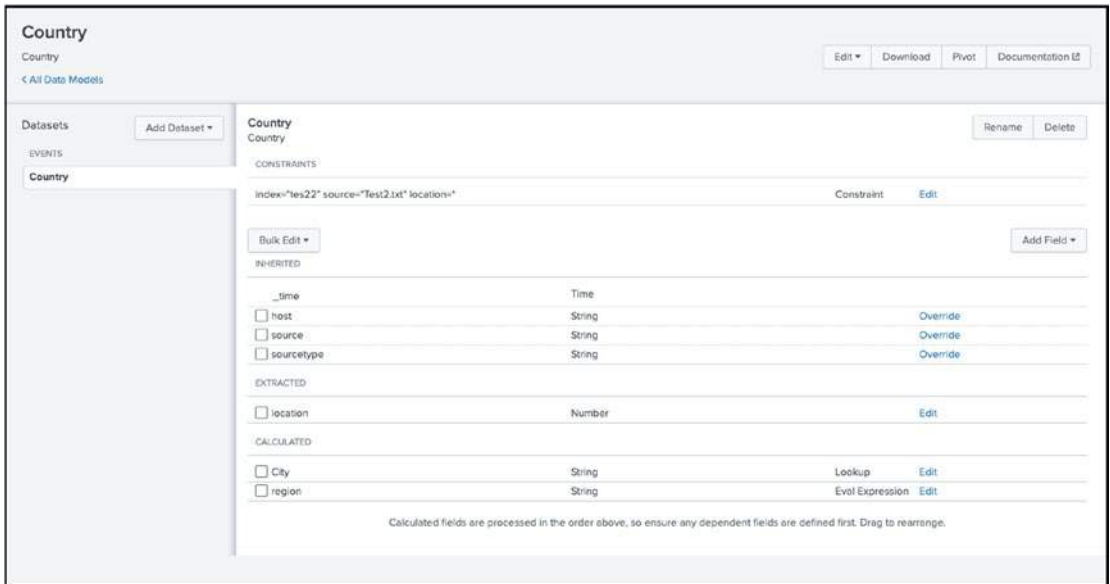
```
case(City in("New york","Chicago","Boston",
"Washington D.C."),"Eastern USA",City in("San
Francisco","Seattle"),"Western USA")
```

These steps are shown in Figure 5-5.



**Figure 5-5.** Evaluate Expression:region

3. Save the changes.
4. After adding locations using the Auto-Extraction States from Lookup and region through Eval Expression, the dataset looks similar to Figure 5-6.



**Figure 5-6.** Dataset Country

5. If you click Pivot, you can see the exact count for the product sales request obtained from the Eastern and Western USA.
6. Select Pivot as the Country dataset.
7. Split the rows as region and column values as \_time.
8. Go to the column chart, and in the range, select All Time.
9. Create a child dataset individually for the Western and the Eastern USA to obtain the product sales requests from each.
10. Click the Country dataset name. Go to Add Dataset, and select Child.
11. In Dataset Name, type **Western USA**. In the search command, type the following command.

```
region="Western USA"
```

Figure 5-7 shows the process of adding a child node named Western USA to the Country dataset.

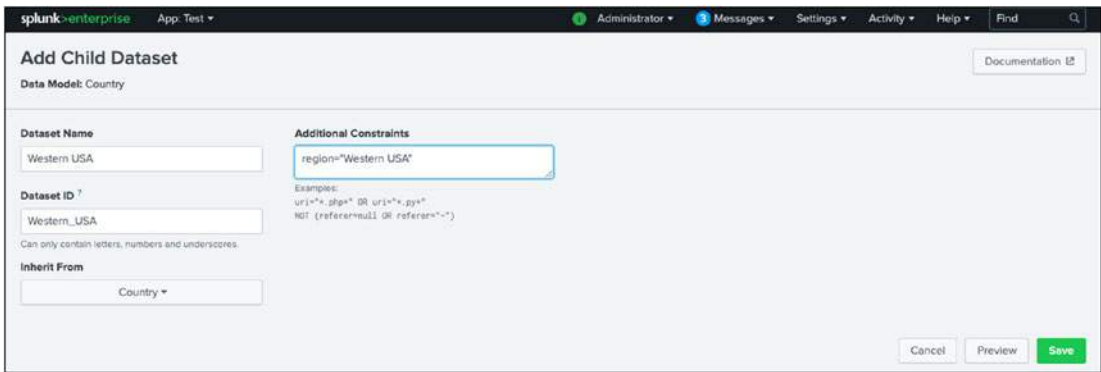


Figure 5-7. Dataset Country:Western USA

Child datasets inherit all datasets from the parent dataset.

12. Similarly, create another child for Eastern USA. Make the dataset name to Eastern USA, and type the following search command. For more help, you can refer to the screenshot in Figure 5-8.

```
region="Eastern USA"
```

Figure 5-8 shows the process of adding a child node named Eastern USA to the Country dataset.

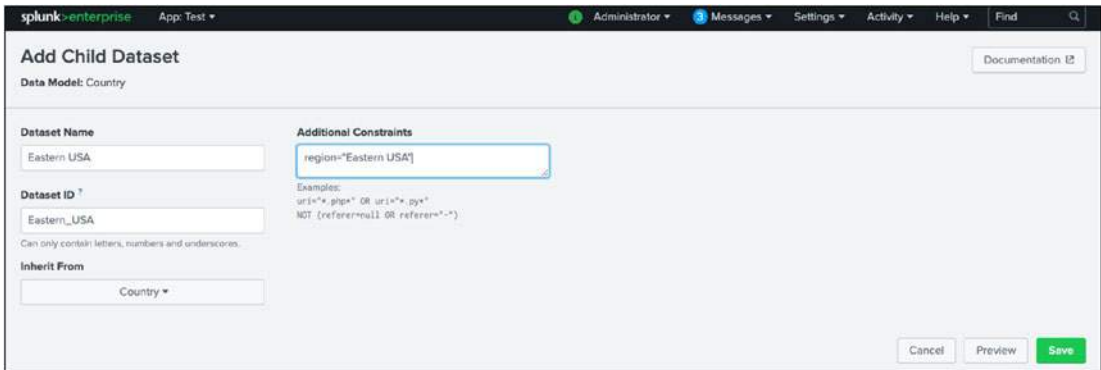
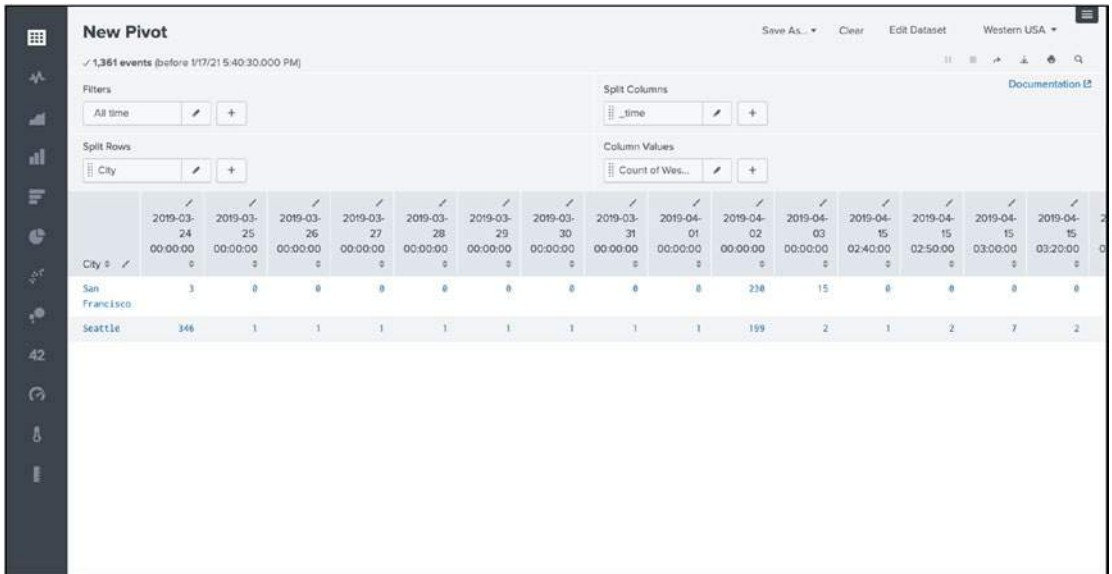


Figure 5-8. Dataset Country:Eastern USA

13. To get product sales requests from cities in the Western USA region, navigate to the Western USA child in Dataset name Country.

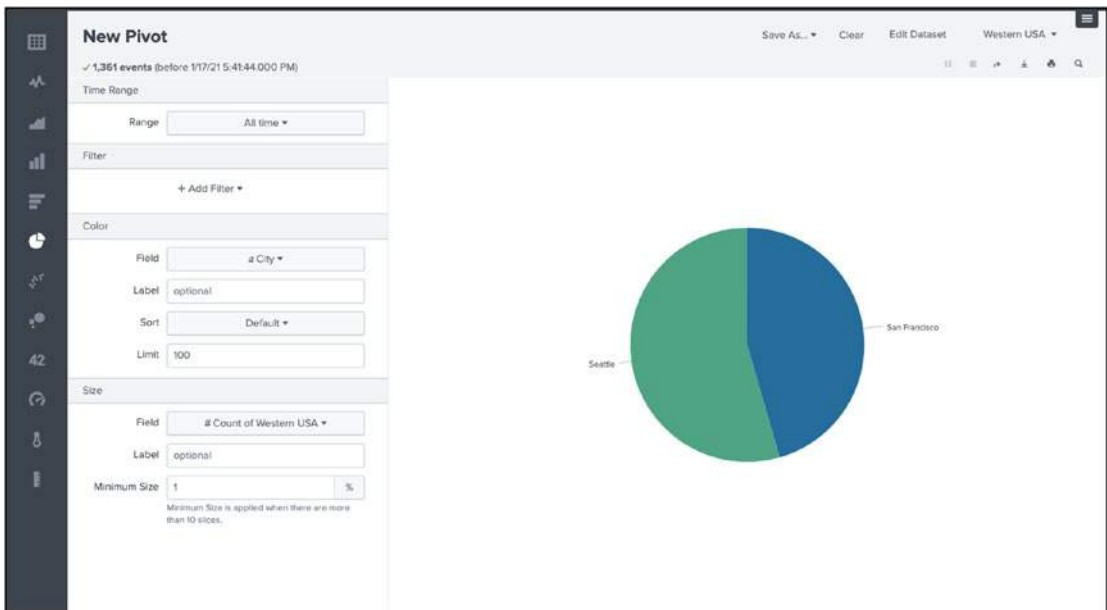
14. Click Pivot for Dataset as Western USA.
15. In Split Rows, select City. In Split Columns, select `_time`.

The screenshot in Figure 5-9 depicts the process of getting product sales requests from cities in the Western USA region.



**Figure 5-9.** Product sales Request Table:Western USA

16. Click Pie chart. In Range, select All Time. You should see a pie chart similar to the one in Figure 5-10.



**Figure 5-10.** Product Sales Request Pie Chart:Western USA

The data model provides a dataset for Pivot.

17. To get product sales from states in the Eastern USA, navigate to Eastern USA child in Dataset Name Country.
18. Click Pivot for Dataset as Eastern USA.
19. In Split Rows, select City. In Column Values, select Count of Eastern USA.

The screenshot in Figure 5-11 depicts the process of getting product sales requests from cities in the Eastern USA region.



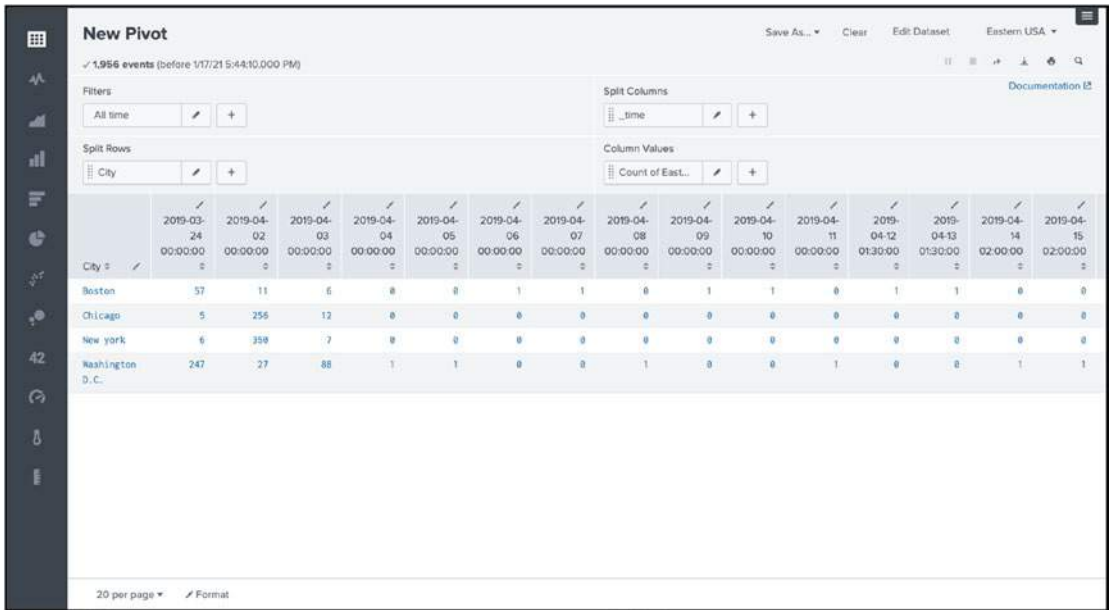


Figure 5-11. Product sales Request Table:Eastern USA

20. Click Pie chart. In Range, select All Time. You should see a pie chart similar to the one in Figure 5-12.

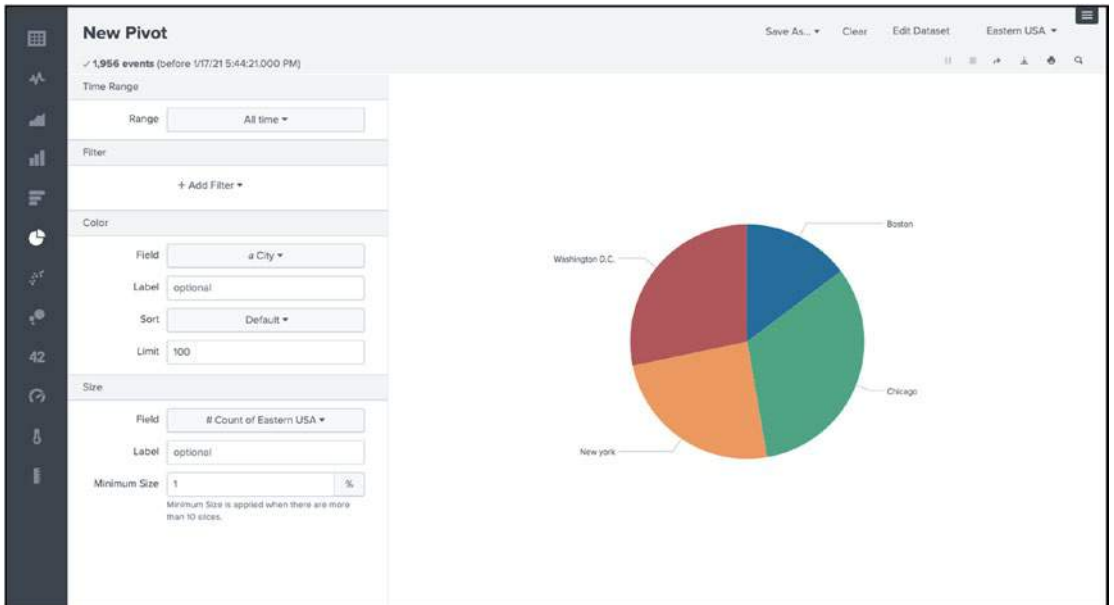


Figure 5-12. Product Sales Request Pie Chart:Eastern USA

Fields in the dataset can be used in Pivot to filter events.

You have completed the datasets and data models. After learning about data set creation and practicing with a command-based application, you now move to event actions in Splunk.

## Event Actions in Splunk

An event action, also known as a workflow, is considered a thoroughly configured knowledge object responsible for starting communication between a field in Splunk Web and other conformable Internet sources. For instance, person A has IP as a field value, and person B has a webpage designed to handle IP requests. Here, B gets the class of IP field, lookup for IP location, and so forth. So, using an event action, you can generate post requests and parse data into a specific URI.

In this section, you launch secondary searches for the IP field from the test.txt source using event actions. You generate a request for the IP field in Splunk Web.

There are three types of workflows in Splunk.

- GET workflow actions
- POST workflow actions
- Search workflow actions

## GET Workflow Actions

The GET workflow action is responsible for passing field values to HTML links, similar to an HREF tag. By clicking a HyperText Transfer Protocol (HTTP) link, the GET request allows you to pass Splunk field value to a specific URI.

## Defining a GET Workflow Action

The following are the steps to create a GET workflow in Splunk.

1. Go to Settings and select Fields.
2. Go to Workflow actions.
3. Click New to open a workflow action.
4. In the workflow action form, provide the following field name values and set an open link in New Window.

```

Label=search_ip
apply only to the following fields=ip
URI=www.google.com/search?ip=$ip$
Link method=get

```

Figure 5-13 shows the field values for getting a workflow action form.

The screenshot shows a configuration form for a workflow action. The fields and their values are as follows:

- Label \***: search\_ip
- Apply only to the following fields**: ip
- Apply only to the following event types**: (empty)
- Show action in**: Both
- Action type \***: link
- Link configuration**:
  - URI \***: www.google.com/search?q=\$ip\$
  - Open link in**: New window
  - Link method**: get

At the bottom right, there are two buttons: "Cancel" and "Save".

**Figure 5-13.**

The workflow action's name cannot have spaces or special characters.

5. Click Save.
6. To check the GET workflow action, go to the Test app, and type the following search command.

```
index="test" source="test.txt"
```

- Click any random event and click Event actions. There, you find search\_ip as a link. Click the link; it performs an HTTP GET request to google.com.

Figure 5-14 shows the field values for a search workflow action form.



Figure 5-14.

## Search Workflow Action

A search workflow action **generates dynamic searches in Splunk**. Instead of writing the entire SPL command for a particular field or determining the result for getting output, you can use a search workflow action for effective search results.

### Defining Search Workflow Action

Here’s an example of creating a search workflow action to get the total count of IP values in the `_time` parameter.

- Go to Settings and select Fields.
- Go to Workflow actions.
- Click New to open a workflow action.
- In the workflow action form, provide the following field name values and set an open link in New Window.

```
Name=ip_count_by_report
Label=$ip$
apply only to the following fields=ip
show action in=Both
```

Action type=Search  
 Search string=index="test"|timechart count by \$ip\$  
 Run in app=test

Figure 5-15 shows the field values for the search workflow action form.

Destination app: test

Name \* ip\_count\_by\_report  
 Enter a unique name without spaces or special characters. This is used for identifying your workflow action later on within Splunk Settings.

Label \* \$ip\$  
**Required field**  
 Enter the label that appears for this action. Optionally, incorporate a field's value by enclosing the field name in dollar signs, e.g. 'Search for ticket number : \$ticketnum\$'.

Apply only to the following fields: ip  
 Specify a comma-separated list of fields that must be present in an event for the workflow action to apply to it. When fields are specified, the workflow action only appears in the field menus for those fields; otherwise it appears in all field menus.

Apply only to the following event types:

Specify a comma-separated list of event types that an event must be associated with for the workflow action to apply to it.

Show action in: Both

Action type \* search

**Search configuration**

Search string \* index="test"|timechart count by \$ip\$  
 Enter the search for this action. Optionally, specify fields as \$fieldname\$, e.g. sourcetype=rails controller=\$controller\$ error="\*".

Run in app: test  
 Choose an app for the search to run in. Defaults to the current app.

Open in view:

Enter the name of a view for the search to open in. Defaults to the current view.

Run search in: New window

**Time range**

Earliest time:

Latest time:

Use the same time range as the search that created the field listing

Cancel Save

Figure 5-15.

The search can run in a new window or the current window in the search workflow action.

5. Click Save.
6. To check the search workflow action, go to the Test app, and type the following search command

```
index=" test" source=" test.txt"
```

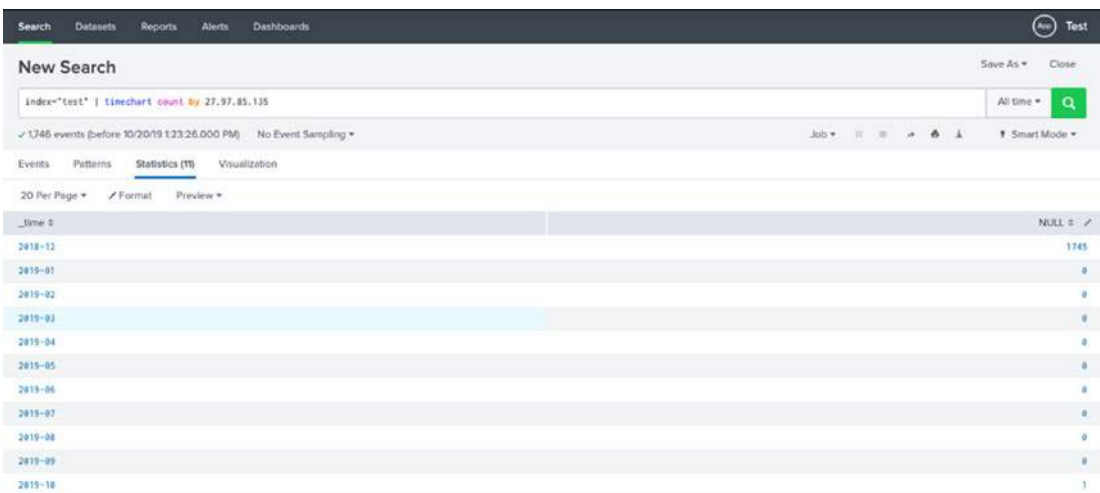
7. Click any random event and then on event actions. Here, you find the IP value as a link that generates a dynamic search in Splunk.

Figure 5-16 shows the field values for the search workflow action form.



**Figure 5-16.**

Figure 5-17 shows the output of the search workflow action that provides the total counts of a particular IP on a time chart.



**Figure 5-17.**

This sums up event actions. Let's now discuss CIM in Splunk.

## Common Information Model in Splunk

A Common Information Model (CIM) is an add-on collection of data models that runs during the search. Every data model in Splunk is a hierarchical dataset. In CIM, the data model comprises tags or a series of field names. They normalize data, using the same field names and event tags to extract from different data sources.

### Defining CIM in Splunk

The following steps show an example of defining CIM.

1. Create a tag named `privileged_location` (the field is `location` and the value is `4`) for the `test2.txt` data source.
2. After creating a tag named `privileged_location`, go to Settings and select Data Models ► Create New.
3. Give Title as `test` and select App Name as `test`.
4. Click Add Dataset and select Root Event.
5. In Dataset Name, type **test**. In Constraints, type the following search command.

```
(index="tes22" source="Test2.txt")  
"tag::location"=privileged_location
```

Figure 5-18 illustrates the process of adding root events in the test dataset.

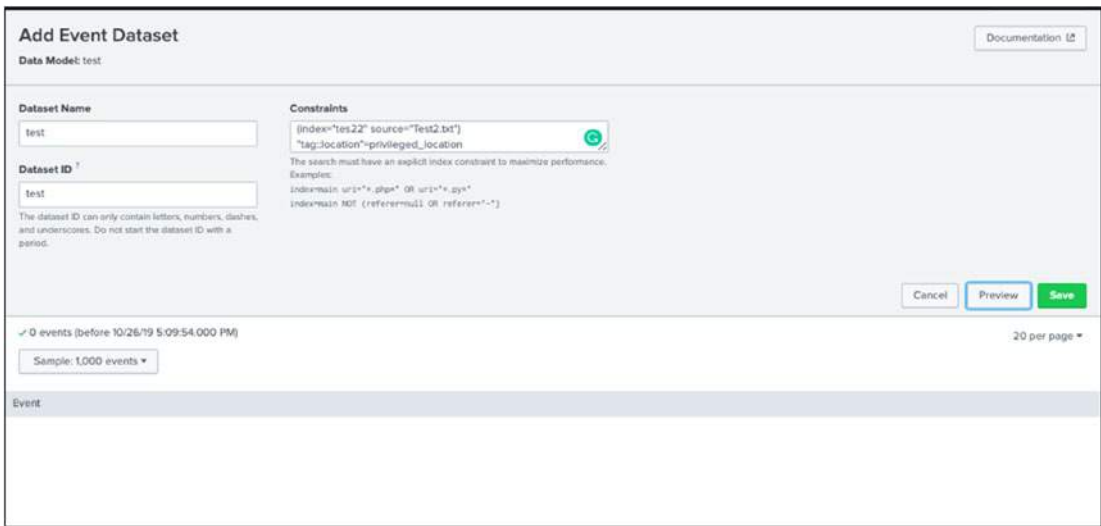


Figure 5-18.

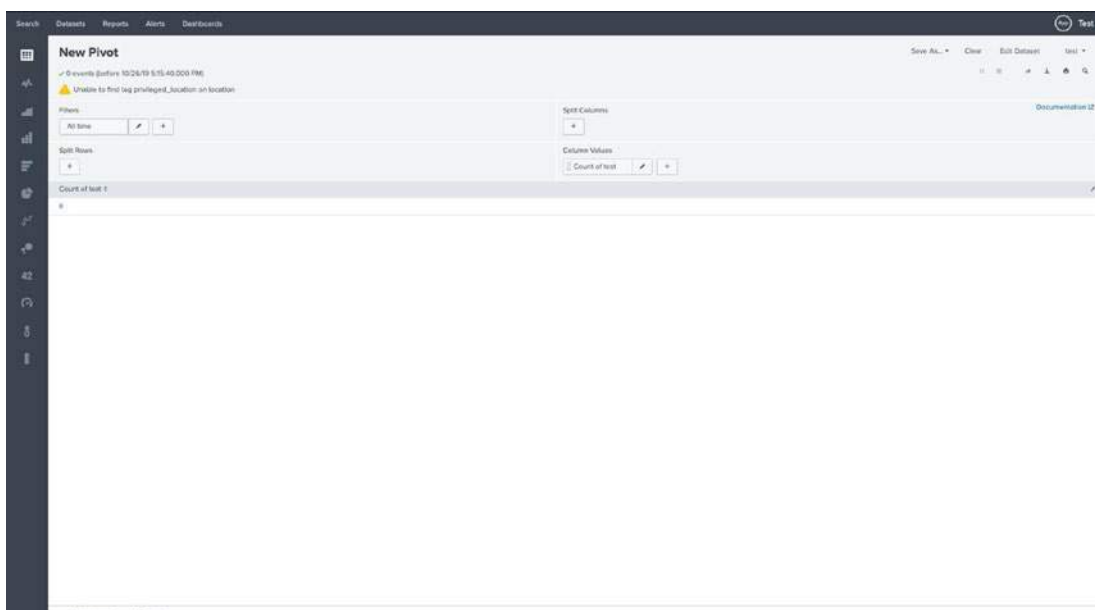
6. Click Save.
7. Click Add Field.
8. Go to Auto-Extracted.
9. Select location, method, status, and unit\_id, as shown in Figure 5-19.



Figure 5-19.

10. Now if you click Pivot, you see the zero count, as shown in Figure 5-20.





**Figure 5-20.**

Since you got zero counts, you need to set up CIM in Splunk. To do this, follow these steps.

1. Go to the Splunk Enterprise home page, locate More Apps, search for CIM, and install it.
2. To set up CIM, go to Apps ► Manage Apps ► Splunk Common Information Model.
3. Go to the web. In the indexes whitelist, select tes22 (see Figure 5-21).

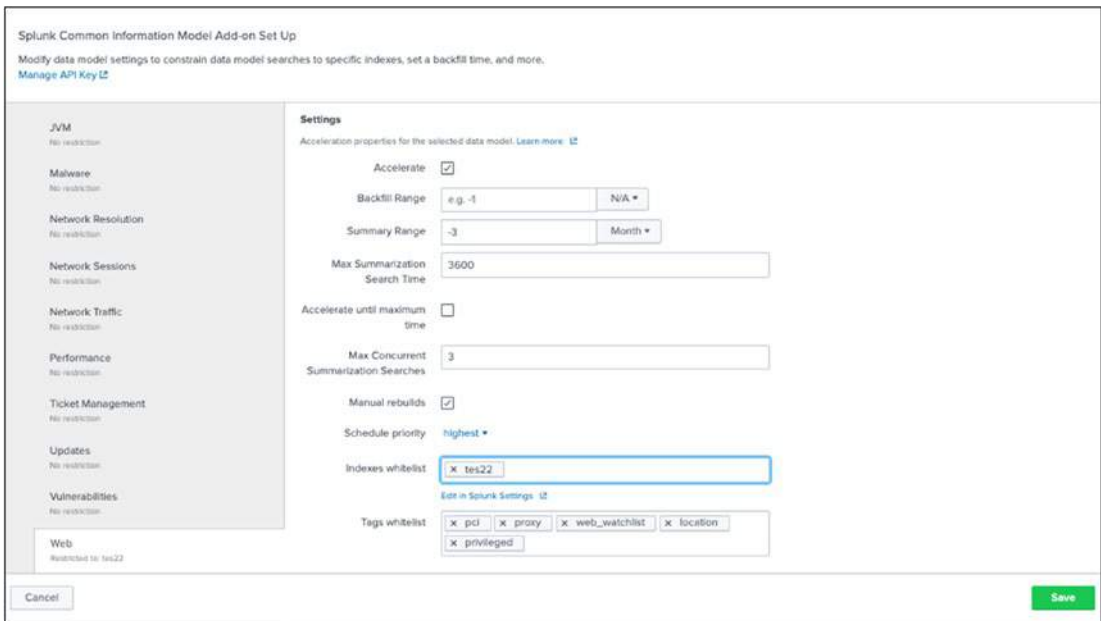


Figure 5-21.

4. Return to the test data model. Refresh the page and run the Pivot model. This time you can see the count. Refer to Figure 5-22 to see the Pivot report.

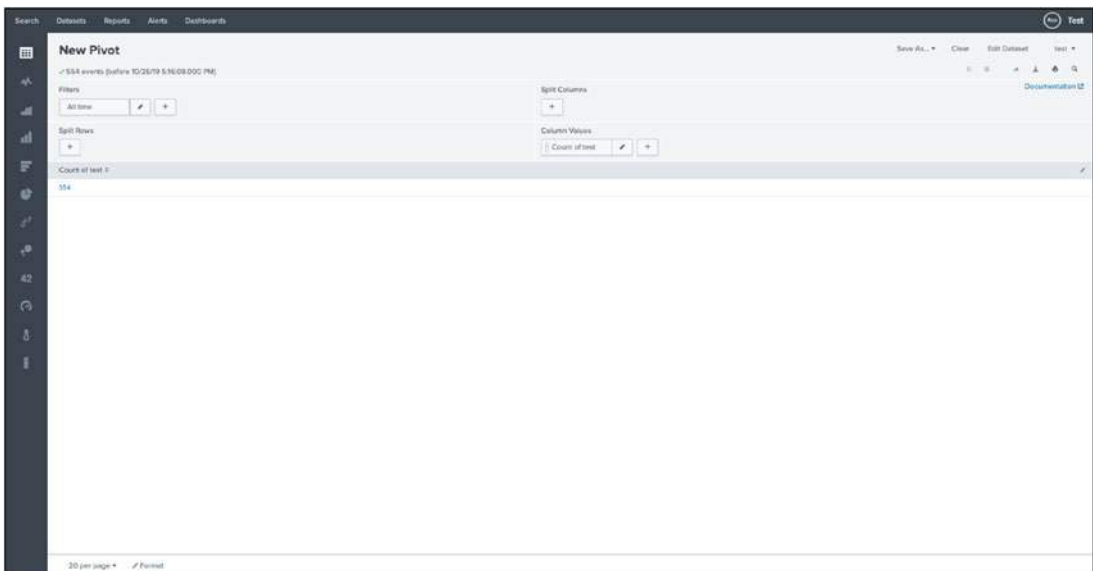


Figure 5-22.

Data models included in CIM are configured with data model acceleration turned off.

You have reached the end of the chapter. I applaud you on successfully learning about datasets, data models, event actions, and CIM.

## Summary

You learned that Splunk commands are simple and easy to understand. They are linked to the HTTP and HTML without additional code or a high-level language and have common execution.

You have learned a hefty portion of workflow actions. Looking at the Splunk Power User exam blueprint, you have become familiar with 10% of Module 9 (data models and Pivot), 10% of Module 8 (event actions), and 10% of Module 10 (CIM models).

The next chapter covers knowledge managers and how to create dashboards in Splunk.

## Multiple-Choice Questions

- A. Select all options that are included as a type of events in a data model.
  - 1. events
  - 2. transactions
  - 3. searches
  - 4. none of the above
- B. What is the relationship between a data model and Pivot?
  - 1. The data model provides a dataset for Pivot.
  - 2. Pivot and data models have no relationship.
  - 3. Pivot and data model are the same things.
  - 4. Pivot provides a dataset for the data model.

- C. Which fields are included in datasets?
  1. auto-extracted
  2. eval expression
  3. lookups
  4. regular expression
  5. geo IP fields
  6. all the above
  7. none of the above
  
- D. Constraints in a dataset are essentially searched terms.
  1. true
  2. false
  
- E. Child datasets inherit all datasets from the parent dataset.
  1. true
  2. false
  
- F. Data models included in CIM are configured with data model acceleration turned on.
  1. true
  2. false
  
- G. Select all options that include a type of workflow action in Splunk.
  1. post
  2. get
  3. search
  4. none of the above

**Answers**

- a. 1, 2, 3
- b. 4
- c. 6
- d. 1
- e. 2
- f. 1
- g. 1, 2, 3

**References**

- <https://docs.splunk.com/Splexicon:Dataset>
- <https://docs.splunk.com/Documentation/Splunk/7.2.4/Alert/CronExpressions>
- [www.youtube.com/watch?v=1IYezUcNGPY&t=1008s](http://www.youtube.com/watch?v=1IYezUcNGPY&t=1008s)
- *Splunk Operational Intelligence* by Josh Diakun (Packt Publishing, 2018)
- <https://docs.splunk.com/Documentation/Splunk/7.2.4/Knowledge/Abouttagsandaliases>

## CHAPTER 6

# Knowledge Managers and Dashboards in Splunk

The previous chapter covered various knowledge objects, software components, commands, and features in Splunk. It dealt with input-output technicalities and provided step-by-step guides. This chapter discusses codes and segments and introduces the manual elements responsible for performing a few of these functions.

This chapter discusses knowledge managers. Programs are written for people to read and machines to execute. Knowledge managers maintain the ease of communication between the machine and the user. You also learn about the two types of dashboards in Splunk and perform assisting commands and operations.

The following topics are covered in this chapter.

- Knowledge managers
- Creating a dashboard in Splunk
- Dynamic form-based dashboards

## Understanding the Knowledge Manager's Role in Splunk

A knowledge manager is a person who provides centralized oversight and maintenance of knowledge objects in Splunk Enterprise. A knowledge manager creates, maintains, and modifies knowledge objects. Knowledge managers can reassign existing knowledge objects to another user and help the stack to tackle day-to-day issues.

Five types of knowledge objects were covered in the previous chapters.

- Data interpretation (fields and field extraction)

- Data classification (event types and transaction)
- Data enrichment (lookups and workflow actions)
- Data normalization (tags and aliases)
- Data models

What exactly does a knowledge manager do? How can a knowledge manager help solve day-to-day issues?

The following describes knowledge managers' responsibilities.

- They declare knowledge objects as global so that the entire organization can access them.
- They make knowledge objects available to apps in Splunk.
- They provide certain users access to a particular knowledge object.
- They manage orphaned knowledge objects.
- They restrict the read/write permission at the app level.

A knowledge manager builds data models for Pivot and ensures that the right person in the organization handles the knowledge objects. They also take care of normalizing a data event.

Now, let's discuss knowledge objects in greater detail.

## Globally Transferring Knowledge Objects

Using Splunk Security, knowledge objects can be declared global to allow every user access to them. For example, the sales, management, and production teams all have access to sales in the United States. This enables the sales team to recognize the sales graph, the management team to understand the organization's sales patterns, and the production team to know which products are profitable. By default, only a Splunk Power User or admin can modify knowledge objects.

To make knowledge objects global, follow these steps.

1. Go to Settings in Splunk Web and select All Configurations.
2. Search for the knowledge object that you want to make global and then click Permissions (see Figure 6-1).

**Object should appear in**

Keep private  
 This app only (test)  
 All apps

**Permissions**

Roles	Read	Write
<b>Everyone</b>	<input type="checkbox"/>	<input type="checkbox"/>
admin	<input type="checkbox"/>	<input type="checkbox"/>
can_delete	<input type="checkbox"/>	<input type="checkbox"/>
power	<input type="checkbox"/>	<input type="checkbox"/>
splunk-system-role	<input type="checkbox"/>	<input type="checkbox"/>
user	<input type="checkbox"/>	<input type="checkbox"/>

**Figure 6-1.** Knowledge Objects Visibility & Access Management

3. To make a knowledge object global, the object should be present in the All Apps option. Under Permission, set the Read and Write roles to Everyone (see Figure 6-2).

**Object should appear in**

Keep private  
 This app only (test)  
 All apps

**Permissions**

Roles	Read	Write
<b>Everyone</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
admin	<input type="checkbox"/>	<input type="checkbox"/>
can_delete	<input type="checkbox"/>	<input type="checkbox"/>
power	<input type="checkbox"/>	<input type="checkbox"/>
splunk-system-role	<input type="checkbox"/>	<input type="checkbox"/>
user	<input type="checkbox"/>	<input type="checkbox"/>

**Figure 6-2.** Knowledge Objects Visibility:Global

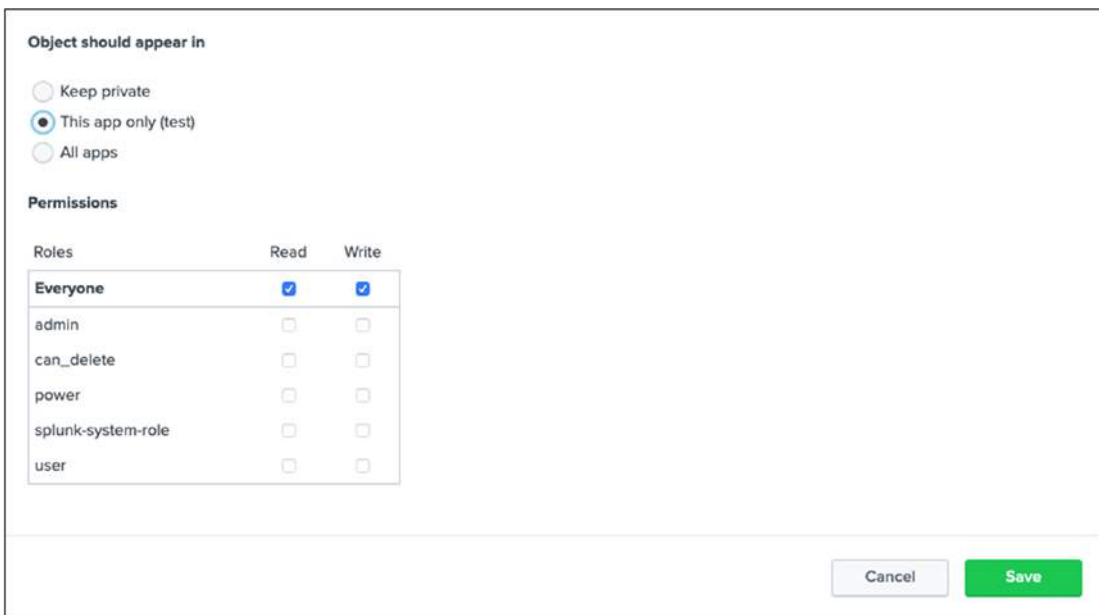


## Enabling Knowledge Object Visibility

The knowledge objects related to sensitive information or hierarchical issues have restricted visibility. Everyone in your organization does not have access. For example, the sales, management, and production teams all have access to data relating to Eastern and Western USA sales. But the organization’s Supply Chain Management (SCM) team does not need this report. You can restrict object visibility to a specified audience.

To make a knowledge object selectively accessible, follow these steps.

1. Go to Settings and select All Configurations.
2. Search for the knowledge object that you want, and then click Permissions. I want Country to be visible only to users of the test app.
3. To make knowledge objects visible after app changes, objects should appear in a particular app. In Permissions, set Read and Write roles to Everyone (see Figure 6-3).



**Figure 6-3.** Knowledge Objects Visibility:test app

## Restricting Read/Write Permissions on an App

Read/write permissions on an app is another knowledge object restriction. If you want users to only read objects and make no edits to them, follow these steps.

1. Go to Settings and select All configurations.
2. Search for the knowledge object and then click Permissions. For example, you Country will be a read-only object.
3. The object must be in All apps.
4. In Permissions, change Read and Write roles to admin and power. Provide read permission to admin, can\_delete, power, Splunk-system-role, and user (see Figure 6-4).

**Object should appear in**

Keep private  
 This app only (test)  
 All apps

**Permissions**

Roles	Read	Write
Everyone	<input type="checkbox"/>	<input type="checkbox"/>
admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
can_delete	<input checked="" type="checkbox"/>	<input type="checkbox"/>
power	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
splunk-system-role	<input checked="" type="checkbox"/>	<input type="checkbox"/>
user	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Figure 6-4.** Knowledge Objects: Access Management

## Orphaned Knowledge Objects

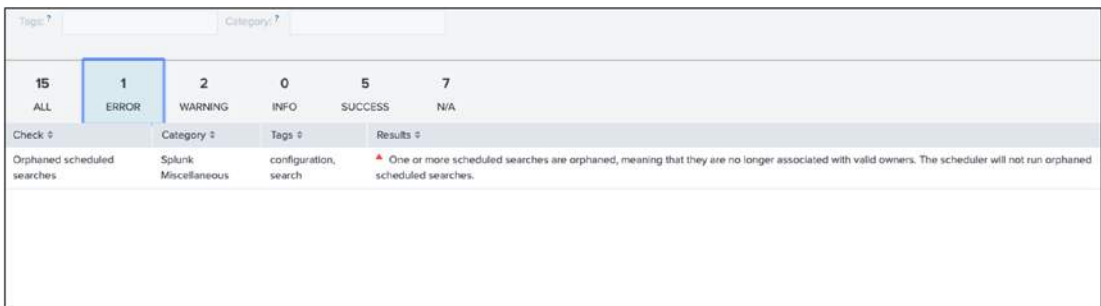
When employees leave an organization, the knowledge objects that were linked to them are deactivated; however, the links to these objects remain. These objects are called *orphaned knowledge objects*. Objects without a legitimate owner are a hindrance in a Splunk environment. And, the scheduled reporter cannot trace and report them because the owner(s) is no longer available. To get around this, Splunk provides various methods to detect orphaned knowledge objects.

- It runs Monitoring Console health checks.
- It uses the Reassign Knowledge Objects page in Settings.
- It reassigns a single knowledge object to another owner.

### Run a Monitoring Console Health Check

If you have admin role access in Splunk Enterprise, you can use the Monitoring Console. It is a health check feature that detects orphaned scheduled searches, reports, and alerts. It tells you the number of knowledge objects that exist in your system. The following steps monitor health checks.

1. Go to Monitoring Console and click Health Check.
2. Click Start.
3. Go to Error, where you find **Orphaned scheduled searches** (see Figure 6-5).



**Figure 6-5.** Orphaned Knowledge Objects

## Using the Reassign Knowledge Objects Page in Settings

The Reassign Knowledge Objects page in Settings is one of the easiest methods for discovering if an orphaned knowledge object exists in a Splunk environment. Refer to the following steps.

1. Select Settings and Go to All configurations.
2. Click Reassign Knowledge Objects.
3. Click Orphaned to filter it from the list.

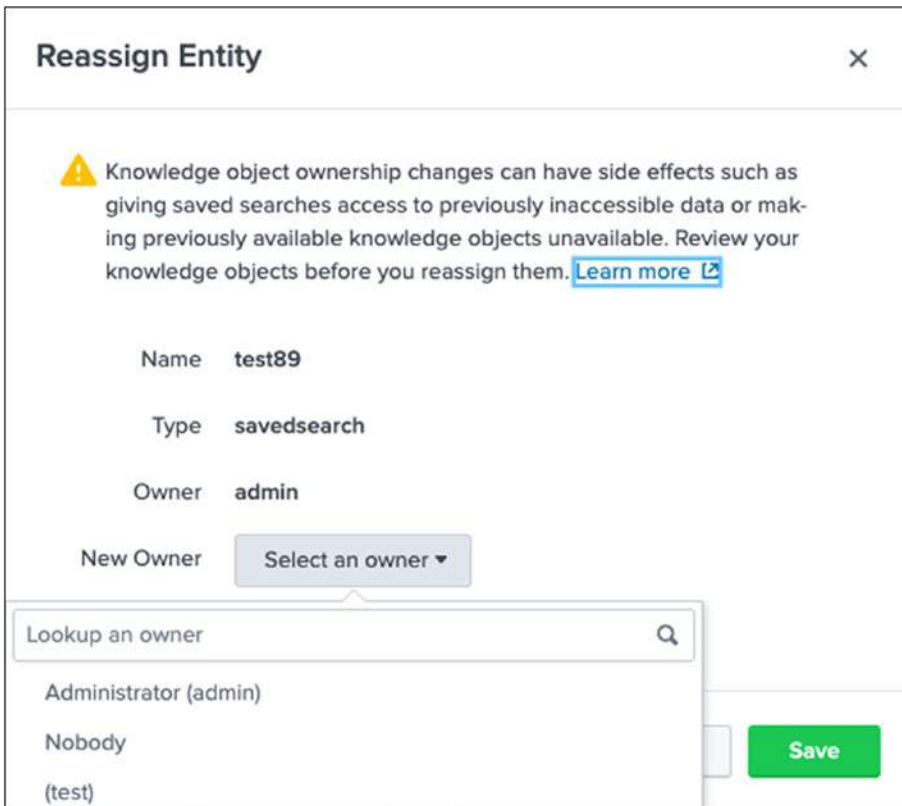
The orphaned button also displays shared orphaned objects.

## Reassigning a Knowledge Object to Another Owner

The Reassign Knowledge Objects page is used for reassigning a knowledge object to a new owner. You can reassign owned and orphaned knowledge objects. Follow these steps to do this.

1. Select Settings and Go to All configurations.
2. Click Reassign Knowledge Objects.
3. Find the object or objects that you want to reassign.

Figure 6-6 shows the Reassign Entry dialog box.



**Figure 6-6.** Reassign Knowledge Objects

Knowledge objects in Splunk are stored in macros.conf, tags.conf, eventtypes.conf, and savedsearches.conf.

At this stage of the chapter, you have already learned about knowledge manager’s roles, and how to regulate a knowledge object’s visibility and restrict its read and write permissions. In the following sections, you learn about dashboards in Splunk.

## Dashboards

Dashboards are designed to easily visualize data in Splunk through quick analyses and information summaries. The dashboard is convenient and straightforward because of the panels that constitute it. Panels include charts, tables, and lists directly linked to the reports.

There are two types of dashboards in Splunk.

- Static real-time dashboards
- Dynamic form-based dashboards

## Static Real-Time Dashboards

The static real-time dashboards display critical information about an organization. Since they are real time, they have higher utility. For example, if you have a manufacturing unit, real-time data is crucial when making significant decisions like reordering inventory stock or getting insights on a production unit. You can also create alerts.

In this section, you learn to create a static, real-time dashboard for the Test2.txt dataset, where the user determines the total number of transactions on his web page, the total Eastern USA sales and total Western USA sales, the status of the HTTP transaction, the HTTP method, and the various product categories.

To create a dashboard for Test2.txt, you need the following 11 reports.

- Total transaction requests on the web page
- Total transaction requests from Western USA
- Total transaction requests from Eastern USA
- Total successful transaction requests on the web page
- Total successful transaction requests from Western USA
- Total successful transaction requests from Eastern USA
- Total sales in Western USA cities
- Total sales in Eastern USA cities
- HTTP status code for the web page
- HTTP method for the web page
- Total transaction requests on the web page for different categories

The solution for creating 11 reports is explained next. If you're already confident with creating a dashboard, try making the reports without using the instructions.

## **Creating a Report in Splunk to Get a Total Transaction Request on the Web Page**

To create this report, the following steps must be observed.

1. Go to Settings.
2. Go to Data models.
3. Go to Country.
4. Select Pivot.
5. Click Pivot and select a single value.
6. Save as a report on total transaction requests.

## **Creating a Report in Splunk to Get a Total Transaction Request from Western USA**

To create this report, observe the following steps.

1. Go to Settings.
2. Go to Data models.
3. Go to Country.
4. Select Pivot.
5. Select Western USA.
6. Click Pivot and select a single value.
7. Save as a report on total transaction requests in Western USA.

## **Creating a Report in Splunk to Get a Total Transaction Request from Eastern USA**

To create this report, observe the following steps.

1. Go to settings.
2. Go to Data models.
3. Go to Country.

4. Select Pivot.
5. Select Eastern USA.
6. Click Pivot and select a single value.
7. Save as a report on total transaction requests in Eastern USA.

## Creating a Report in Splunk to Get a Successful Transaction Request on the Web Page

To create this report, observe the following steps.

1. Create a data model like the one in Chapter 5 for successful transactions. The only change is to add a constraint field status of 200 to the existing constraint for the Country data model. Refer to the following input constraint field.

```
index="tes22" sourcetype=test2 location=* status=200
```

2. Create a new data model named Country2 and save the report for successful transaction requests from the Eastern and Western USA.

Save the report for successful transaction requests from Eastern USA and Western USA as described.

## Creating a Total Sales Report for Western US Cities

To create this report, observe the following steps.

1. Go to Settings.
2. Go to Data models.
3. Go to Country.
4. Select Pivot.
5. Select WesternUSA2.
6. Click Pivot, and in Split rows, select City.
7. Click Pie Chart.
8. Save as a report on total sales in Western USA cities.



## Creating a Total Sales Report for Eastern US Cities

To create this report, observe the following steps.

1. Go to Settings.
2. Go to Data models.
3. Go to Country.
4. Select Pivot.
5. Select EasternUSA2.
6. Click Pivot, and in Split rows, select City.
7. Click Pie Chart.
8. Save as a report on total sales in Eastern USA cities.

## Creating Report for an HTTP Status Code

To create this report, observe the following steps.

1. Go to Settings.
2. Select Searches, Reports, and Alerts.
3. Select New Report.
4. In Title, type **Status**. In search, type the following query.

```
index="tes22" sourcetype=test2 |timechart count by status
```

5. Click Save.

## Creating a Report for an HTTP Method

To create this report, observe the following steps.

1. Go to Settings.
2. Select Searches, Reports, and Alerts.
3. Select New Report.
4. Name the report **Method**.

5. In the search, type a query similar to the following.

```
index="tes22" sourcetype=test2 |timechart count by method
```

6. Click Save to secure the report.
7. Click Save.

## Creating Report to Get a Total Transaction Request for Different Categories

To create this report, observe the following steps.

1. Go to Settings.
2. Select Searches, Reports, and Alerts.
3. Select New Report.
4. Name the report **Categories**.
5. In the search, enter a query similar to the following.

```
index="tes22" sourcetype=test2 | timechart count by Category
```

6. Click Save.

## Creating a Dashboard

Let's now create a dashboard named Sales for the application test. It consists of the total transaction requests in the web page report, the total transaction requests from the Western USA report, the total transaction requests from the Eastern USA report, the total successful transaction requests in the web page report, the total successful transaction requests from the Western USA report, the total successful transaction requests from the Eastern USA report, the total sales in Western USA cities report, the total sales in Eastern USA cities report, the HTTP status code for the web page report, the HTTP method for the web page report, and the total transaction requests on the web page report in different categories.

1. Click Dashboards, which is located on the navigation bar at the top of the screen.

2. Click Create New Dashboard to fill out the dashboard properties.
3. In Dashboard Title, type **Sales**. In Dashboard Permissions, select Shared in App.

Figure 6-7 shows the Dashboard panel.

**Save As Dashboard Panel** X

Dashboard  New  Existing

Dashboard Title

Dashboard ID ?   
Can only contain letters, numbers and underscores.

Dashboard Description

Dashboard Permissions  Private  Shared in App

---

Panel Title

Panel Powered By ? **Q Inline Search**

Drilldown ? **No action**

Panel Content

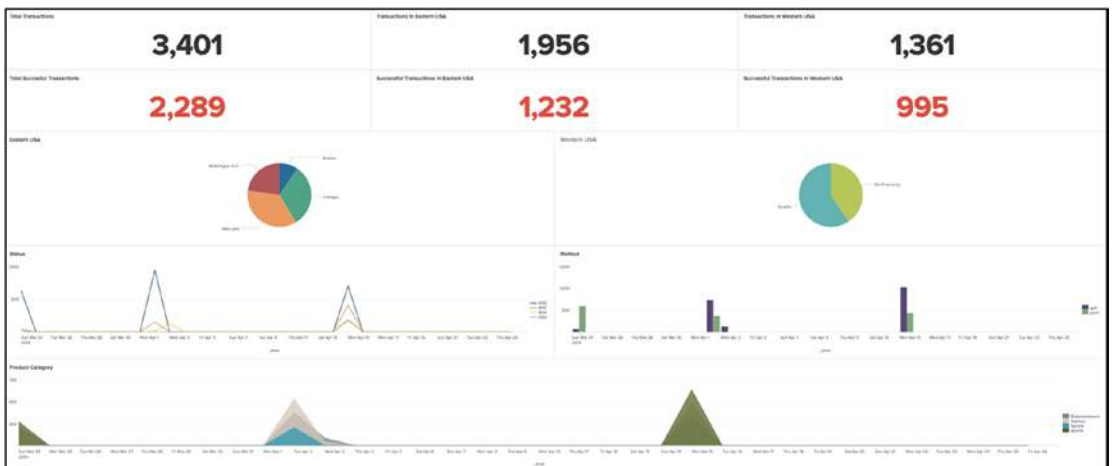
**Figure 6-7.** Create a Dashboard:Sales

## Adding a Report to a Dashboard

To add a dashboard in Splunk, observe the following steps.

1. On the navigation bar at the top of the page, click Dashboards.
2. Select the Sales dashboard.
3. Click Edit and Click **Add panel** at the top of the page. Select **New from the report**.
4. Select all 11 reports and add them to the Sales dashboard.

I created the dashboard shown in Figure 6-8.



**Figure 6-8.** Sales Dashboard

5. After adding all reports and making considerable changes to the dashboard, click Save.

You can generate code from your dashboard by going to it and clicking Edit ► Source. I generated XML code for my Sales dashboard, and I uploaded it to GitHub at <https://github.com/deepmehta/splunk-certification-guide/blob/main/ch6/Sales.xml>.

The last section of this chapter discusses dynamic form-based dashboards and their input fields.

## Dynamic Form-based Dashboards

Dynamic form-based dashboards allow users to filter data without moving to a new page. It enables you to add input fields, such as radio buttons, link lists, time modifiers, and drop-downs. This type of dashboard is used for troubleshooting, getting insights into data, and attaining business intelligence.

First, let's discuss adding radio buttons using XML.

### Adding a Radio Button Using XML

A radio button is a control element that allows the user to choose any one of the available options and to filter data based on category. It is added through a dynamic form-based dashboard. For example, if I want to see all the events that come under the Sports category, the simplest way to add radio buttons is to use source code.

```
<input type="radio" token="token_name">
-----
-----
</input>
```

Design a dynamic form-based dashboard with radio buttons to filter data based on a category value. Include a column chart that displays the event count by category.

Try to do it on your own. If you need a reference, use the following code. It is followed by a screenshot of the dashboard.

```
<form>
<label>Test2</label>
<description>Test2</description>
<fieldset submitButton="true" autoRun="true">
  <input type="radio" token="Category">
    <label>Category</label>
    <choice value="*">All</choice>
  <search>
    <query>index="tes22" sourcetype=test2 category=*</query>
  </search>
```

```

    <fieldForLabel>Category</fieldForLabel>
    <fieldForValue>Category</fieldForValue>
    <default>*</default>
  </input>
</fieldset>
<row>
  <panel>
    <table>
      <title>index="tes22" sourcetype=test2 category=$Category$</title>
      <search>
        <query>index="tes22" sourcetype=test2 category=$Category$ </query>
        <earliest></earliest>
        <latest></latest>
      </search>
      <option name="showPager">true</option>
    </table>
  </panel>
</row>
<row>
  <panel>
    <chart>
      <title>Count by Category</title>
      <search>
        <query>index="tes22" sourcetype=test2 |stats count by category </
        query>
        <earliest></earliest>
        <latest></latest>
      </search>
      <option name="charting.chart">Column Chart</option>
    </chart>
  </panel>
</row>
</form>

```



```

<form>
  <label>Test2 Copy</label>
  <description>Test2</description>
  <fieldset submitButton="true" autoRun="true">
    <input type="radio" token="Category">
      <label>Category</label>
      <choice value="*">All</choice>
      <search>
        <query>index="tes22" sourcetype=test2 category=*</query>
      </search>
      <fieldForLabel>Category</fieldForLabel>
      <fieldForValue>Category</fieldForValue>
      <default>*</default>
    </input>
    <input type="time" token="field1">
      <label></label>
      <default>
        <earliest></earliest>
        <latest></latest>
      </default>
    </input>
  </fieldset>
</row>
<panel>
  <table>
    <title>index="tes22" sourcetype=test2 category=$Category$ |timechart
      count by Category</title>
    <search>
      <query>index="tes22" sourcetype=test2 category=$Category$
        |timechart count by Category</query>
      <earliest></earliest>
      <latest></latest>
    </search>
    <option name="showPager">true</option>
  </table>

```



```

</panel>
</row>
<row>
  <panel>
    <chart>
      <title>Count by Category</title>
      <search>
        <query>index="tes22" sourcetype=test2 |stats count by category </
          query>
        <earliest></earliest>
        <latest></latest>
      </search>
      <option name="charting.chart">Column Chart</option>
    </chart>
  </panel>
</row>
</form>

```

Figure 6-10 shows the dynamic form-based dashboard with radio buttons and the time modifier. It also has a column chart with a category-based event count.



Figure 6-10.

## Adding a Drop-Down Menu Using XML

The third control element is the drop-down that allows the user to choose only one of the available options and filter data based on category, time, and a defined status. It is done through the dynamic form-based dashboard.

For example, if I want to see all events that have that fall in a Sports category, on March 23, 2019 with a 200 status, the simplest way to do this is to add a drop-down with the following source code.

```
<input type="dropdown" token="token_name" searchWhenChanged="true">
-----
-----
</input>
```

Design a dynamic form-based dashboard with radio buttons, a time modifier, and a drop-down menu to filter data based on category value, time, and status. Include a column chart that displays the event count by category.

Try to do it on your own. If you need a reference, use the following code. It is followed by the screenshot of the dashboard.

```
<form>
  <label>Test3 Copy</label>
  <description>Test3</description>
  <fieldset submitButton="true" autoRun="true">
    <input type="radio" token="Category">
      <label>Category</label>
      <choice value="*">All</choice>
      <search>
        <query>index="tes22" sourcetype=test2 Category=* status=*</query>
      </search>
      <fieldForLabel>Category</fieldForLabel>
      <fieldForValue>Category</fieldForValue>
      <default>*</default>
    </input>
    <input type="time" token="field1">
      <label></label>
      <default>
```

```

    <earliest></earliest>
    <latest></latest>
</default>
</input>
<input type="dropdown" token="status" searchWhenChanged="true">
  <label>Status</label>
  <choice value="200">200</choice>
  <choice value="400">400</choice>
  <choice value="404">404</choice>
  <choice value="500">500</choice>
</input>
</fieldset>
<row>
  <panel>
    <table>
      <title>index="tes22" sourcetype=test2 Category=$Category$ $status$
        |table status and Category</title>
      <search>
        <query>iindex="tes22" sourcetype=test2 Category=$Category$
          $status$ |timechart count by Category</query>
        <earliest></earliest>
        <latest></latest>
      </search>
      <option name="showPager">true</option>
    </table>
  </panel>
</row>
<row>
  <panel>
    <chart>
      <title>Count by Category</title>
      <search>
        <query>index="tes22" sourcetype=test2 |stats count by Category
          </query>
        <earliest></earliest>

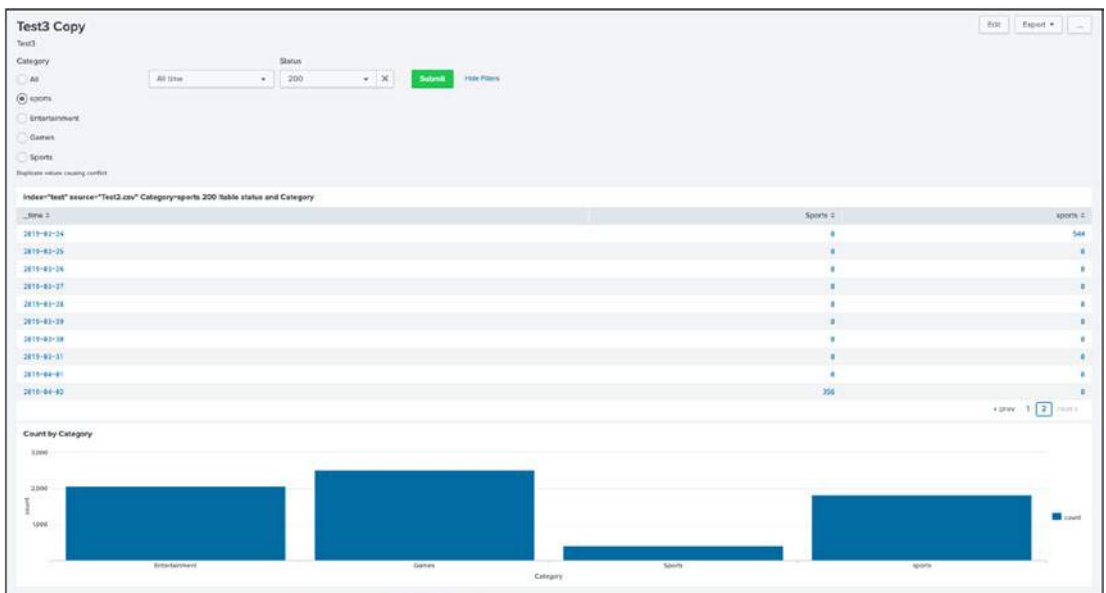
```

```

    <latest></latest>
  </search>
  <option name="charting.chart">Column Chart</option>
</chart>
</panel>
</row>
</form>

```

Figure 6-11 shows the dynamic form-based dashboard with the radio buttons, the time modifier, and the drop-down. It also has a column chart with a category-based event count.



**Figure 6-11.**

Finally, let's add a link list knowledge object.

## Adding a Link List Using XML

The next control element is the link list. It allows the user to choose only one of the available options and filters data based on category, time frame, status, and method.

For example, I want to see all events in the Sports category on March 23, 2019, with a 200 status, and the GET method.

The simplest way to add a link list is to use the following source code.

```
<input type="link" token="token_name" searchWhenChanged="true">
-----
</input>
```

Using the radio button, time modifier, drop-down, and link list, design a dynamic form-based dashboard that filters data based on a category value, time, status, and method. Include a column chart that displays the event count by category.

Try to do it on your own. If you need a reference, use the following code. It is followed by a screenshot of the dashboard.

```
<form>
<label>Test3 Copy</label>
<description>Test3</description>
<fieldset submitButton="true" autoRun="true">
  <input type="radio" token="Category">
    <label>Category</label>
    <choice value="*">All</choice>
    <search>
      <query>index="tes22" sourcetype=test2 category=* status=*</query>
    </search>
    <fieldForLabel>Category</fieldForLabel>
    <fieldForValue>Category</fieldForValue>
    <default>*</default>
  </input>
  <input type="link" token="method" searchWhenChanged="true">
    <label>method</label>
    <choice value="get">get</choice>
    <choice value="post">post</choice>
  </input>
  <input type="time" token="field1">
    <label></label>
    <default>
      <earliest></earliest>
      <latest></latest>
    </default>
```

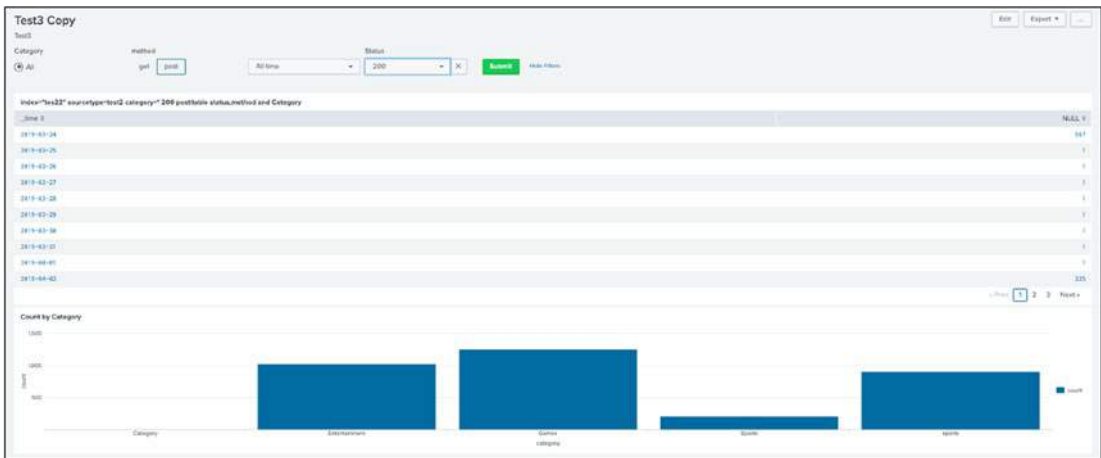
```

</input>
<input type="dropdown" token="status" searchWhenChanged="true">
  <label>Status</label>
  <choice value="200">200</choice>
  <choice value="400">400</choice>
  <choice value="404">404</choice>
  <choice value="500">500</choice>
</input>
</fieldset>
<row>
  <panel>
    <table>
      <title>index="tes22" sourcetype=test2 category=$Category$ $status$
        $method$|table status,method and Category</title>
      <search>
        <query>index="tes22" sourcetype=test2 category=$Category$ $status$
          $method$ |timechart count by Category</query>
        <earliest></earliest>
        <latest></latest>
      </search>
      <option name="showPager">true</option>
    </table>
  </panel>
</row>
<row>
  <panel>
    <chart>
      <title>Count by Category</title>
      <search>
        <query>index="tes22" sourcetype=test2 |stats count by category
          </query>
        <earliest></earliest>
        <latest></latest>
      </search>
      <option name="charting.chart">Column Chart</option>
    </chart>
  </panel>
</row>

```

```
</chart>  
</panel>  
</row>  
</form>
```

Figure 6-12 shows the dynamic form-based dashboard with the radio button, the time modifier, the drop-down, and the link list. It also has a column chart with a category-based event count.



**Figure 6-12.**

You have completed adding dynamic-based dashboards using XML. Next, you learn how to input data through the user interface.

## Using the User Interface for Input

Using the user interface (UI) for input is simplified when done through a dynamic form-based dashboard. Follow these steps to add input through the UI.

1. Go to the Dashboard page and select the dashboard.
2. Click Edit.
3. Go to Add Input and select the desired input.
4. Select Text.

- Click the Edit button. In Label, enter **Category**. In Token, enter **Category**, and select Search on Change (see Figure 6-13).

The screenshot shows the configuration window for a Text widget in Splunk. On the left, a sidebar lists widget types: Text (selected), Radio, Dropdown, Checkbox, Multiselect, Link List, and Time. The main area is titled 'General' and contains the following fields:

- Label:** A text input field containing the word 'Category'.
- Search on Change:** A checkbox that is checked.
- Token Options:** A section containing several input fields:
  - Token ?**: A text input field containing 'Category'.
  - Default ?**: An empty text input field.
  - Initial Value ?**: An empty text input field.
  - Token Prefix ?**: An empty text input field.
  - Token Suffix ?**: An empty text input field.

At the bottom of the window, there are two buttons: 'Cancel' on the left and 'Apply' on the right.

**Figure 6-13.**

The radio button, the drop-down, the checkbox, and the link list can all be added using the same UI steps. Adding input using UI code is comprehensible and accommodating enough to allow beginners to operate them. However, the XML code is comparatively more complex.

I want you to commend yourself for completing the chapter. You learned about the knowledge managers and functions performed on knowledge objects and the dashboard and its types. Practice the commands to get a better understanding of their functions.



## Summary

Splunk is a user-friendly platform for searching, processing, and monitoring big data. It encourages easier code and commands in a hierarchical arrangement of its elements and objects. This chapter furthered the discussion on the knowledge objects. It also talked about the knowledge managers, the people responsible for maintaining knowledge objects. You analyzed the dashboard and saw how it easily provides big data visualization. You also learned how to input data using the UI and XML.

The next chapter is a Splunk User and Power User mock test.

You have covered 5% of Module 5 (knowledge manager roles) and 12% of Module 6 (static real-time dashboard and the dynamic form-based dashboards).

## Multiple-Choice Questions

- A. In the Reassign Knowledge Objects page in Settings, you can view shared knowledge objects.
  - 1. true
  - 2. false
  
- B. Knowledge objects are stored in\_\_\_\_\_. (Select all options that apply.)
  - 1. macros.conf
  - 2. tags.conf
  - 3. eventtypes.conf
  - 4. savedsearches.conf
  - 5. props.conf
  - 6. transform.conf
  
- C. A knowledge manager's role includes (Select all options that apply.)
  - 1. Granting users access to appropriate knowledge objects
  - 2. Building data models for Pivot

3. Normalizing data
  4. None of the above
- D. Which is a type of knowledge object? (Select all options that apply.)
1. Data interpretation (fields and field extraction )
  2. Data classification (event types and transaction)
  3. Data enrichment (lookups and workflow actions)
  4. Data normalization (tags and aliases)
  5. Data models
  6. None of the above
- E. Select the control elements of a dynamic form-based dashboard.
1. radio button
  2. time modifier
  3. link list
  4. drop-down
  5. aliases
  6. None of the above

**Answers**

- a. 1
- b. 1, 2, 3, 4
- c. 1, 2, 3
- d. 1, 2, 3, 4, 5
- e. 1, 2, 3, 4

## References

- <https://docs.splunk.com/Documentation/Splunk/7.2.6/AdvancedDev/AdvancedDashboard>
- <https://docs.splunk.com/Documentation/Splunk/7.2.6/AdvancedDev/AdvancedDashboard>
- [https://docs.splunk.com/Documentation/Splunk/6.1.1/Viz/FormEditor#Add\\_a\\_time\\_input\\_to\\_a\\_for](https://docs.splunk.com/Documentation/Splunk/6.1.1/Viz/FormEditor#Add_a_time_input_to_a_for)

## CHAPTER 7

# Splunk User/Power User Exam Set

In Chapter 6, you learned why a Splunk knowledge manager's role is important. You also learned how to create an advanced dashboard in Splunk. This chapter features multiple-choice questions (MCQ) that are useful in preparing for Splunk power user and user certification. In this chapter, you get a better idea of the kinds of questions that appear on these certification exams.

## Questions

- A. What are the main components of Splunk?
  1. Splunk forwarder, Splunk indexer, and Splunk search head
  2. Splunk indexer, Splunk heavy forwarder, and Splunk search head
  3. Splunk indexer, Splunk deployment manager, and Splunk forwarder
  4. Splunk heavy forwarder, Splunk deployment manager, and Splunk indexer
- B. When you install Splunk on a stand-alone machine, which components are on it?
  1. input data
  2. parser
  3. indexer
  4. all of the above

- C. When you install the Splunk Enterprise free 60-day trial version, what is your daily index size?
  - 1. 100 MB
  - 2. 500 MB
  - 3. 750 MB
  - 4. 1000 MB
  
- D. A \_\_\_\_\_ displays statistical trends over time.
  - 1. chart value
  - 2. time series
  - 3. stats value
  - 4. table value
  
- E. The \_\_\_\_\_ command can display any series of data you want to plot.
  - 1. chart
  - 2. stat
  - 3. time chart
  - 4. both 1 and 2
  
- F. Which are filtering commands in Splunk?
  - 1. where, dedup, and head
  - 2. dedup, head, and tail
  - 3. where, dedup, and tail
  - 4. all of the above
  
- G. A transaction is a grouping command.
  - 1. true
  - 2. false

- H. Splunk extracts default fields when adding data.
1. true
  2. false
- I. Delimiters are used for \_\_\_\_ data.
1. structured
  2. unstructured
- J. Which default fields does Splunk extract when adding data?
1. host, datatype, and source type
  2. host, source type, and source
  3. 1 and 2
  4. none of the above
- K. When you run macros in Splunk, you need to rewrite the entire command.
1. true
  2. false
- L. Select the option that consists of lookup types.
1. CSV lookup, KV Store lookup, and external lookup
  2. CSV lookup, geospatial lookup, and KV Store lookup
  3. KV Store lookup, external lookup, and geospatial lookup
  4. all of the above
- M. Tags are used for \_\_\_\_ pairs.
1. field/value
  2. key/value
  3. 1 and 2
  4. Field/Key

- N. As a knowledge manager, you are responsible for building data models that provide Pivot.
1. true
  2. false
- O. Data model acceleration doesn't use automatically generated summaries to speed up completion time in Pivot.
1. true
  2. false
- P. To create an alert in Splunk using .conf, you must edit \_\_\_\_.
1. savedsearches.conf
  2. props.conf
  3. alert.conf
  4. transformation.conf
- Q. You can access a tag in Splunk using \_\_\_\_\_. (Select all options that apply.)
1. ::
  2. =
  3. \*
  4. &
- R. Field aliases appear in all fields and interesting fields if they appear at least \_\_\_\_.
1. 10%
  2. 80%
  3. 40%
  4. 20%

- S. Select all options that include the type of events in the data model as a dataset.
1. events
  2. transactions
  3. searches
  4. none of the above

**Answers**

- A. 1
- B. 4
- C. 2
- D. 2
- E. 1
- F. 4
- G. 1
- H. 1
- I. 1
- J. 2
- K. 2
- L. 4
- M. 3
- N. 1
- O. 2
- P. 1
- Q. 1, 2, 3
- R. 4
- S. 1, 2, 3



## Summary

In this chapter, you answered multiple-choice questions that may be useful for Splunk power user and user certification. You have come to the end of Module 1, which covers the chapters on Splunk power user and user certification. Next, you start learning about Splunk admin certifications.

# **PART II**

# **Splunk Data Administration and System Administration**

## CHAPTER 8

# Splunk Licenses, Indexes, and Role Management

This chapter introduces you to Splunk buckets, licenses, and user role management.

The following topics are covered in this chapter.

- Buckets in Splunk
- journal.gz, .tsidx, and Bloom filter
- Splunk licenses
- Managing Splunk licenses
- User management

## Buckets

Buckets are directories of raw and indexed data. It has two main properties: maximum data and maximum time. Both properties need to be configured to run buckets smoothly. There are three types of buckets.

- **Hot buckets:** After data is parsed into Splunk, it goes through a license master, and the event is written in a hot bucket. Put simply, it is the bucket where data is written. Hot buckets are searchable. A hot bucket switches to a warm bucket when the maximum size is reached or Splunk is restarted. An index may have multiple hot buckets open at a time.

- **Warm buckets and cold buckets:** Warm buckets are searchable but are not actively written. Warm buckets are identified on “\_time”. Warm buckets are rolled to cold buckets when a particular index exceeds a limit. After a particular time, cold buckets are rolled to freezing or archived, depending on the index policy.
- **Freezing:** This is also known as *data expiration*. It is the oldest bucket of data. It is deleted from an index when the index’s size is full or the bucket’s age exceeds the specified time limit.

The `indexes.conf` file manages index policies such as data expiration and data thresholds.

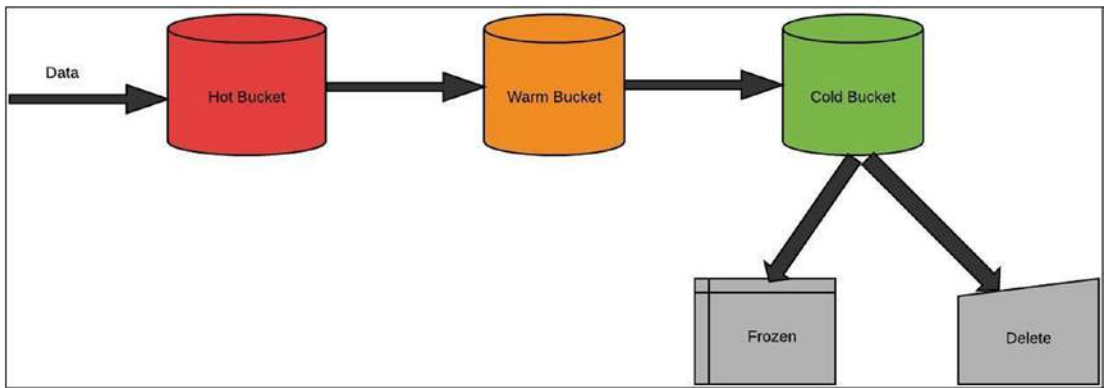
## How Does a Bucket Work?

When you ingest data in a Splunk platform through monitor input, scripted input, or data upload, it is indexed. Splunk saves it in directories called *buckets*. According to data policy definitions, the first bucket is the hot bucket, where the data stays for some time. It is a read-write bucket.

A hot bucket rolls into a warm bucket in either cases: The `splunkd` (a service that accesses, processes, indexes streaming data and handles search requests) is restarted, the index size exceeds the limit, the timespan of bucket is too large, hot bucket has no receiving data in a while or there is an increase in bucket’s metadata. Subsequently then, it creates a new directory named “`db_[newest_time]_[oldest_time]_[ID]`”. It is named on “\_time” to know the events’ time range in a particular bucket.

Time in a warm bucket is expressed as an epoch. A warm bucket is generally read-only and rolls to the cold bucket in either cases if the index limit is reached or created too many of them. In this rollover from warm bucket to cold bucket, the entire bucket directory is copied to a cold bucket path. Ideally, the oldest index would be switched first.

Cold buckets allow you to read data and then either delete the data or roll it over to a frozen bucket, as defined in the policy. In the frozen bucket, the index cannot be read and you cannot write data, only archive it. To read frozen data, it needs to be moved to a cold bucket. The oldest bucket is deleted from the index when the index’s maximum size is reached, or the bucket age reaches the limit. A frozen path can be configured. This flow is represented in the diagram in Figure 8-1.



**Figure 8-1.** *Bucket Life Cycle*

Hot and warm buckets should use faster storage, such as SSD. Cold buckets can use slower storage, such as SANS/NAS.

## How Search Is Performed in Buckets

When you search indexed data, Splunk gives results in reverse. Let's suppose that you want to search data in a recently added event. Splunk directly looks in hot buckets because they are the buckets that have been recently written to. So, it is easy to find recent results.

Similarly, if you search for the keyword *kite*, Splunk checks the hot bucket first. If the word is not found, Splunk displays the timestamp of the event “kite” and the event's actual time coming to the bucket. However, it then looks for the event in the warm bucket in the reverse chronology of `_time` and, if not found, moves to the cold bucket, searching in the same manner. Therefore, in Splunk, the search mainly occurs by checking the recently added events through `_time`, digging into the bucket only when the keywords match. Henceforth, if a “kite” event is newly added and another “kite” event is in the warm or cold bucket, Splunk shows the event in the hot bucket first, followed by the event in the warm or cold bucket.

Let's now discuss the three commands used by the buckets to perform a search: `journal.gz`, `.tsidx`, and Bloom filters.

## Understanding journal.gz, .tsidx, and Bloom Filters

To understand the workings of a search, you need to know the three components that buckets use to perform a search.

- **journal.gz:** When you ingest data in Splunk, it is saved in hot buckets. Inside hot buckets, journal.gz plays a crucial role in saving data. The raw data is compressed and divided into slices. A slice can be up to 128 KB of uncompressed data.
- **.tsidx** stands for *time-series index file*. Each event is given a unique ID within a bucket. An event is segmented into terms and saved in lexicographical (sorted alphabetically) order. Each term is tracked in its .tsidx and a binary search can determine if the term is there or not. If the term is found, it can be scanned linearly to check whether data is there. A lexicon table points it to the address where the raw data is stored. Together, journal.gz and the .tsidx make a bucket.
- **Bloom filters** decrease an indexer's time to retrieve events from an index. A Bloom filter plays a crucial role in predicting whether an event is present in the index or not. Bloom filters run at the index level. For example, if you run a search command with the keyword *error*, a Bloom filter filters all buckets that don't contain that keyword in the lexicon table. Splunk Enterprise saves you time by searching .tsidx files within a specified bucket where the search content is available.

Now you know about buckets and their components. But how does the Splunk search function work?

## How Do Search Functions Work?

Suppose that you want to search for events with the keyword *kite*. When you write a command in the SPL to search for these events using the hash function, it generates a hash key for *kite*. It then searches the hash key on buckets based on `_time` using a Bloom filter. If the Bloom filter determines that the keyword is present in a particular index, it uses the lexical table generated by the time-series index file and seeks the address where the raw data is stored (journal.gz). Together, journal.gz, the Bloom filter, and .tsidx make the search work more efficiently.

Let's now move on to discuss Splunk licenses.

# Splunk Licenses

When you input data in Splunk, the indexer indexes it and stores it on the disk. The Splunk license determines the data ingestion limit. Each Splunk Enterprise instance needs a license that specifies the rules on the amount of data it can index in a day. There are various types of Splunk licenses.

- **Splunk Enterprise license**

The “standard” Splunk Enterprise license specifies the type of data that is indexed, available for purchase, or configured.

- **No-Enforcement license**

The standard Splunk Enterprise license has a maximum daily indexing volume in which you get a violation warning if it is exceeded. If you receive more than five warnings in a month, you violate your license. This may result in the Splunk search head being disabled. However, in no-enforcement, even if you violate the license, your search head won't be disabled.

- **Enterprise Trial license**

The Enterprise Trial license allows maximum indexing of up to 500 MB/day. It expires after 60 days, and you are asked to switch to the standard Splunk Enterprise license or the Free license.

- **Sales Trial license**

The Enterprise Trial license expires after 60 days and has an indexing capacity of 500 MB/day. If you have a pilot project that needs indexing capacity greater than 500 MB/day, you can directly contact the Splunk sales team to get a license.

- **Dev/Test license**

Splunk provides access to its Dev/Test license to operate in a non-production environment.

- **Free license**

The Free license in Splunk has an indexing capacity of up to 500 MB/day. With this license, you cannot perform distributed searches, TCP/HTTP forwarding, alerts, user management, LDAP, or scripted authentication.

- **Splunk for Industrial IoT Licenses**

Splunk provides a special license for industrial IoT that provides access to sets of specifically designed apps.

- **Forwarder License**

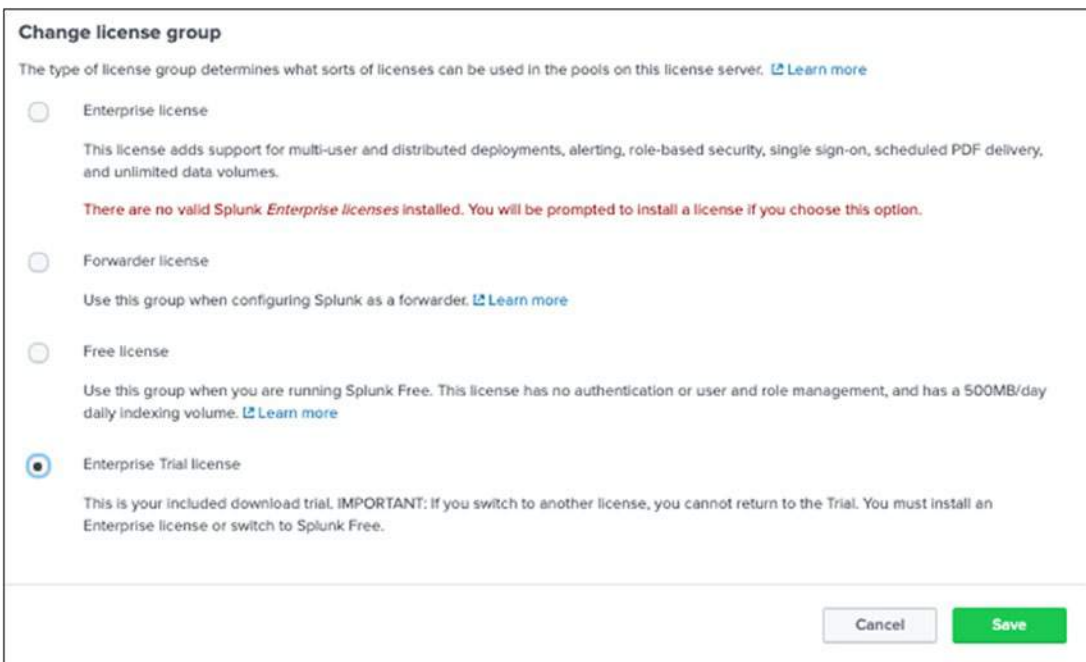
The Forwarder license allows forwarding of unlimited data. Unlike a free license, it enables authentication. You need not buy an extra license because it is included in Splunk.

## Changing a License Group in Splunk

To move from trial to Enterprise license, follow these steps.

1. Go to Settings and select Licensing.
2. Select the License Group that you want to implement.

A sample screen image is shown in Figure 8-2.



**Figure 8-2.** Splunk License Group

Management is the soul of the undisturbed working of any program or software. Let's look at how Splunk licenses are managed.



# Managing Splunk Licenses

Splunk licenses are based on the maximum amount of data you can input. For example, the Splunk Enterprise Trial license group provides 500 MB/day licensing capacity. All the Splunk Enterprise instances need a license according to usage and the type of data being input. The following describes the types of data.

- **Event data** is the “raw\_data” that the indexer inserts into the indexing pipeline.
- **Metrics data** counts as 150 bytes each. It draws from the same license quota as event data.
- **Summary indexing** lets you run fast searches over large datasets by spreading out the cost of a computationally expensive report over time.

License metering occurs in the indexing phase.

Splunk licenses manage the work in two categories.

- License master
- License slave

## License Masters and Slaves

In Splunk, the license master and the license slave work in synchronization. In your enterprise instance, there is one license master instance and all other instances are slave nodes. Cluster master, indexer, deployers, deployment server, Monitoring Console, etc. All other instances are generally slave nodes managed by license master. The role of license master is to manage the license usage of your Enterprise instance.

### License Master

The license master in Splunk manages license access from a central location when there are multiple instances. The license master allocates licensing capacity and manages the license usage for all instances. You simply set one Splunk instance as license master and the remaining are its license slaves. To configure an instance as a license master using Splunk Web, refer to the following steps.

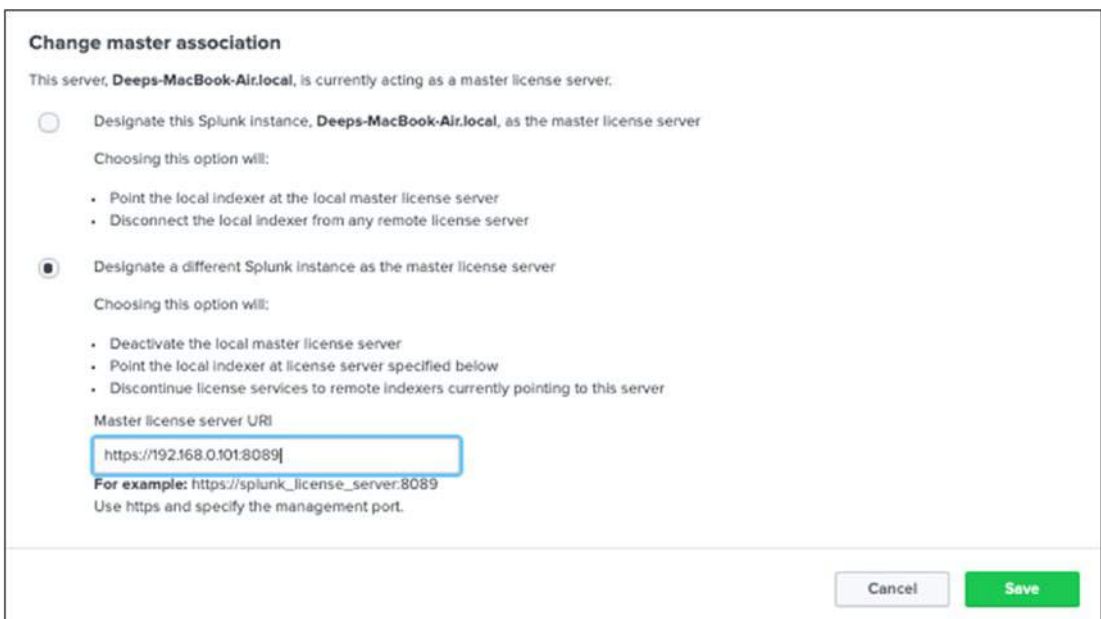
1. Go to Settings and select Licensing.
2. Designate the license server type as master.

## License Slave

When you configure one Splunk instance as the license master, the remaining instances need to be license slaves. However, if you have a single instance, that node acts as the license master without a license slave. To configure an instance as a license slave, follow these steps.

1. Go to Settings and select Licensing.
2. Designate the license server type as slave.
3. Specify which master it should report to. Provide an IP address or hostname along with the Splunk management port (the default is 8089).
4. Click Save and restart the Splunk instance.

Figure 8-3 shows changing the license type.



**Figure 8-3.** License Master

If you have a dedicated license master in your Enterprise instance, the number of slaves is the only factor that needs to be considered. You can have low CPU cores, memory, and disk size.

The next section deals with the process of adding a license in Splunk.

## Adding a License in Splunk

License addition is very important for the long-term implementation of the Splunk software. Suppose you have an Enterprise Trial license and want to switch to a standard Splunk Enterprise license. Similarly, suppose you have an indexing license that allows up to 500 GB/day but your usage is increasing and you need more. You can add more licenses.

To add a license in Splunk, follow these steps.

1. Go to Settings and select Licensing.
2. Select Add License.
3. To install a license, upload a license file (license files end with .license).
4. Click Install.

Figure 8-4 shows how to add a new license in Splunk.



The screenshot shows a dialog box titled "Add new license". It contains the following text: "Learn more about your license options at the [licensing section](#) on splunk.com." Below this, it says "To install a license, upload a license file here (license files end with .license):". There is a file selection area with a "Choose file" button and the text "No file chosen". Below that, it says "Or, copy & paste the license XML directly...". At the bottom right, there are two buttons: "Cancel" and "Install".

**Figure 8-4.** Adding a License

Splunk licenses are saved in `$SPLUNK_HOME/etc/licenses`.

## License Pooling

License pooling allows licenses to be subdivided and assigned into groups of indexers. A license size is allocated to the master node, and the license master manages the license pool and the slave nodes' access to the volume. If you have a Splunk standalone instance, it automatically becomes a licensed master, residing in the Splunk Enterprise stack. It has a default license pool named **auto\_generated\_pool\_enterprise**.

## Creating a License Pool

To create a license pool in Splunk Web, refer to the following.

1. Go to Settings and select Licensing.
2. Click Add Pool at the bottom of the page.
3. Specify a name for the pool.
4. Set the size allocation. Allocation is valuable. It can be the entire amount of indexing volume available in the stack; it depends on your priorities and requirements.
5. Specify the indexers that can access the pool. A particular indexer can have access to a particular pool, or a group of indexers may have access to a particular pool.

You cannot create a licensed pool in the Splunk Enterprise Trial license.

The index forms a very important part of the Splunk environment. It is the building block of the entire software. So, managing it efficiently is crucial. The following section discusses the management of the indexes in Splunk.

## Managing Indexes in Splunk

In Splunk, the indexer processes data and saves it to the main index by default. But, you can create and specify indexes for other data input. The index consists of files and directories, also called *buckets*. Buckets change per defined rules.

Along with the main index, Splunk also has preconfigured internal indexes.

- **main** is the default index. All external data is stored in this index.
- **\_internal** includes internal logs and matrices.
- **\_audit** includes events related to auditing, user search history, and so forth.
- **\_introspection** tracks system performance and usage resources.
- **\_the fishbucket** includes various checkpoints for file monitoring.
- **Summary** is the default index for the summary indexing system.

The following are types of indexes.

- **Events indexes** impose a basic structure and allow all types of events to include metrics data.
- **Metrics indexes** are a highly structured format and impose high latency demands related to metrics data. Putting metrics data into metrics indexes rather than events indexes leads to better performance and less index storage use. Metrics data is counted at a fixed 150 bytes per event.

## Creating an Index in Splunk

Creating an index segregates incoming databases, which helps the Splunk admin bring the right team for the operation. For example, if I created an index named security to collect all events related to my organization's safety, it would help the company's security team. There are several steps to creating an index in Splunk.

### Creating an Index Using Splunk Web

To create an index using the Splunk Web, take the following steps.

1. Go to Splunk Web and select Settings.
2. Go to Indexes and click New.

3. Enter the following information to create a new index.
  - a. The name of your index.
  - b. Enter HOME\_PATH. The default path is `$$SPLUNK_DB/<index_name>/db`
  - c. Enter COLD\_PATH. The default path is `$$SPLUNK_DB/<index_name>/colddb`.
  - d. Enter THAWED\_PATH. The default path is `$$SPLUNK_DB/<index_name>/thaweddb`.
  - e. Enable/Disable integrity check.
  - f. Enter the maximum size of your index. The default is 500,000 MB.
  - g. You can also configure the size of each index bucket according to your needs or else select auto.
  - h. If you want your data to be archived, enter the path in the frozen path.
  - i. Select the app for which you want to create the index.
  - j. For storage optimization configure `tsidx` policy.

Figure 8-5 shows the screen for creating a new index in a Splunk instance.

## New Index ✕

**General Settings**

Index Name   
Set index name (e.g., INDEX\_NAME). Search using Index=INDEX\_NAME.

Index Data Type  Events  Metrics  
The type of data to store (event-based or metrics).

Home Path   
Hot/warm db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/db).

Cold Path   
Cold db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/colddb).

Thawed Path   
Thawed/resurrected db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/thaweddb).

Data Integrity Check  Enable  Disable  
Enable this if you want Splunk to compute hashes on every slice of your data for the purpose of data integrity.

Max Size of Entire Index  GB ▾  
Maximum target size of entire index.

Max Size of Hot/Warm/Cold Bucket  GB ▾  
Maximum target size of buckets. Enter 'auto\_high\_volume' for high-volume indexes.

Frozen Path   
Frozen bucket archive path. Set this if you want Splunk to automatically archive frozen buckets.

App Search & Reporting ▾

**Storage Optimization**

Tsidx Retention Policy  Enable Reduction  Disable Reduction  
**Warning:** Do not enable reduction without understanding the full implications. It is extremely difficult to rebuild reduced buckets. [Learn More](#)

Reduce tsidx files older than  Days ▾  
Age is determined by the latest event in a bucket.

**Figure 8-5.** Create Index:test89

## Creating an Index Using a Splunk Configuration File

To create an index using a configuration file, edit the `indexes.conf` file located in `$Splunk_HOME/etc/system/local`. Refer to the following code block.

```
[security_test]
homePath=<path>
coldPath=<path>
thawedPath=<path>
enableDataIntegrityControl=0|1
enableTsidxReduction=0|1
maxTotalDataSizeMB=<size>
```

1. In `homePath`, enter the path where you want to store your buckets in Splunk. The default path is `$SPLUNK_DB/<index_name>/db`.
2. In `coldPath`, enter the path where you want to store your cold buckets in Splunk. The default path is `$SPLUNK_DB/<index_name>/colddb`.
3. In the thawed patch, enter the path where you want to store your archived buckets in Splunk. The default path is `$SPLUNK_DB/<index_name>/thaweddb`.
4. Enable Data Integrity Control to configure integrity checks in Splunk.
5. Enable Tsidx Reduction for storage optimization. Configure according to your needs.
6. In Max Total DataSize MB, configure the maximum size of your buckets. Please enter the size in MB.

## Creating an Index Using Splunk CLI

To create an index using Splunk CLI, refer to the following steps.

```
splunk add index test <homepath> -coldPath <coldpath>
  -thawedPath <thawedpath> -enableDataIntegrityControl <0|1>
  -enableTsidxReduction <0|1> -maxTotalSizeMB<size>
```



1. In `-homePath`, enter the path where you want to store your buckets in Splunk. The default path is `$$SPLUNK_DB/<index_name>/db`.
2. In `-coldPath`, enter the path where you want to store your cold buckets in Splunk. The default path is `$$SPLUNK_DB/<index_name>/colddb`.
3. In `-thawedPath`, enter the path where you want to store your archived buckets in Splunk. The default path is `$$SPLUNK_DB/<index_name>/thaweddb`.
4. In `-enableDataIntegrityControl`, configure for integrity checks in Splunk.
5. Configure `-enableTsidxReduction`, which covers storage optimization, according to your needs.
6. `-maxTotalDataSizeMB` configures the maximum size of your buckets. Please enter the size in MB.

You've now wrapped up this section. The last section of this chapter discusses user management and software privileges.

## User Management

User management covers the privileges that a user has within Splunk. It determines the roles and working structure of the Splunk admin. There are various user management tasks that a Splunk admin needs to handle.

- Adding native users
- Defining role inheritance and role capabilities
- Role index search options

### Adding a Native User

Adding a native user to Splunk is a simple process. To add native user in Splunk (web), refer to these instructions.

1. Go to Settings and select Users.
2. Click New User.
3. Provide a username and password (required).
4. Provide the user’s full name and email address (defaults to none).
5. Enter your time zone (defaults to search head time zone).
6. Enter the default app (defaults to home).
7. Enter the role (defaults to user).

Figure 8-6 is a screenshot of adding a native user to Splunk.

The screenshot shows the 'Create User' form in Splunk. The form is titled 'Create User' and has a close button (X) in the top right corner. The fields are as follows:

- Name: test
- Full name: test
- Email address: test@gmail.com
- Set password: .....
- Confirm password: .....

Below the password fields, there is a message: 'Password must contain at least 8 characters' with a green checkmark and the number 8.

The Time zone dropdown is set to '(GMT-05:00) Eastern Time (US & Canada)'. The Default app dropdown is set to 'test (Test)'. The Assign roles section has two columns: 'Available Item(s)' and 'Selected Item(s)'. The 'Available Item(s)' column contains: can\_delete, power, splunk-system-role, and user. The 'Selected Item(s)' column contains: admin, power, and user. There are 'add all' and 'remove all' buttons between the columns. At the bottom of the form, there are two checkboxes: 'Create a role for this user' (unchecked) and 'Require password change on first login' (checked). The 'Save' button is green, and the 'Cancel' button is white with a grey border.

**Figure 8-6.** Adding a native user

Password in Splunk is stored in \$SPLUNK\_HOME/etc/passwd.Passwd file in Splunk is Encrypted.

## Defining Role Inheritance and Role Capabilities

A new user can inherit capabilities and index access from another user. If a new user inherits roles and capabilities from the parent user, you cannot disable it. The following steps define role inheritance.

1. Go to Settings and select Users.
2. Select Admin (whichever role you want).
3. Go to Admin (whichever user you want).
4. Go to Inheritance and select whichever roles you want to inherit.

Figure 8-7 shows adding inheritance in Splunk.

**Create User** [X]

Name: test

Full name: test

Email address: test@gmail.com

Set password: [masked]

Confirm password: [masked]

Password must contain at least <sup>?</sup>  
 ✓ 8 characters

Time zone <sup>?</sup>: (GMT-05:00) Eastern Time (US & Canada) ▼

Default app <sup>?</sup>: test (Test) ▼

Assign roles <sup>?</sup>:

Available item(s)	add all ▶	Selected item(s)	◀ remove all
admin		admin	
can_delete		power	
power		user	
splunk-system-role			
user			

Create a role for this user:

Require password change on first login:

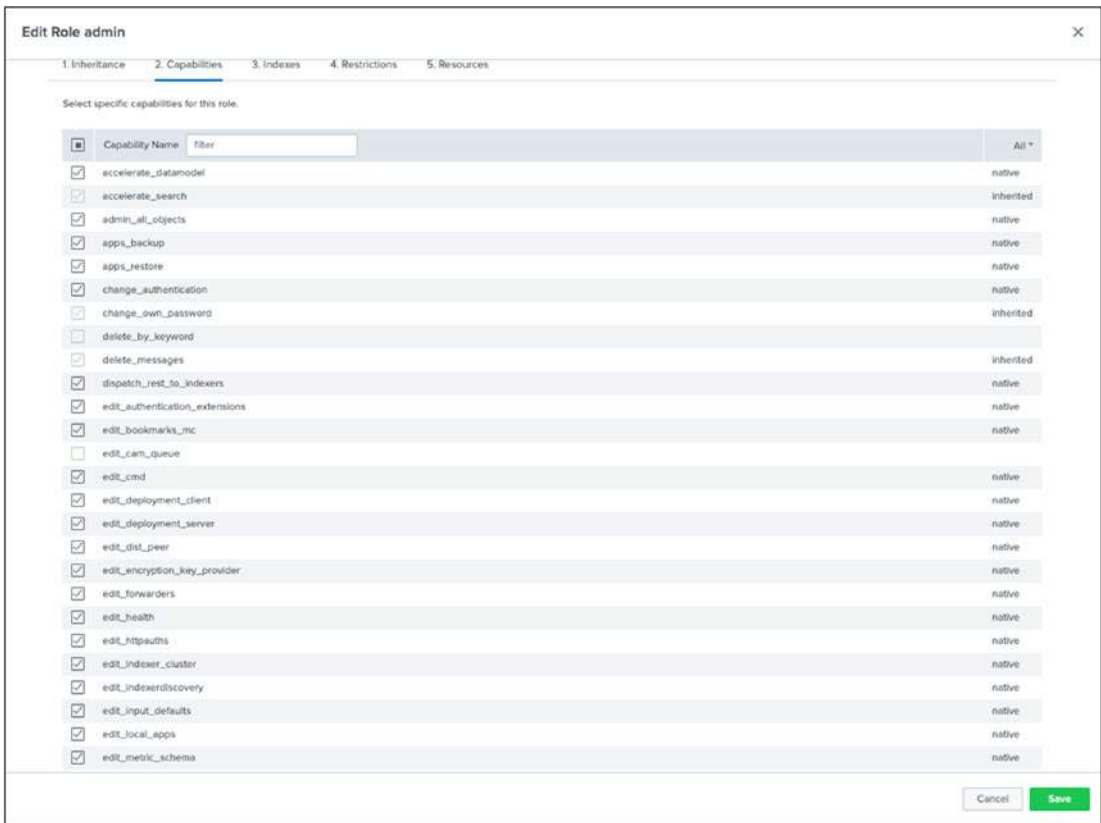
Cancel Save

**Figure 8-7.** *efining role Inheritance to Splunk Users*

Role inheritance can inherit capabilities and index access. The following steps add role capabilities in Splunk.

1. Go to Settings and select Roles.
2. Go to Admin (select whichever user you want).
3. Go to Capabilities and select Assign or Remove Capabilities.

Figure 8-8 is a screenshot of adding role capabilities to Splunk.



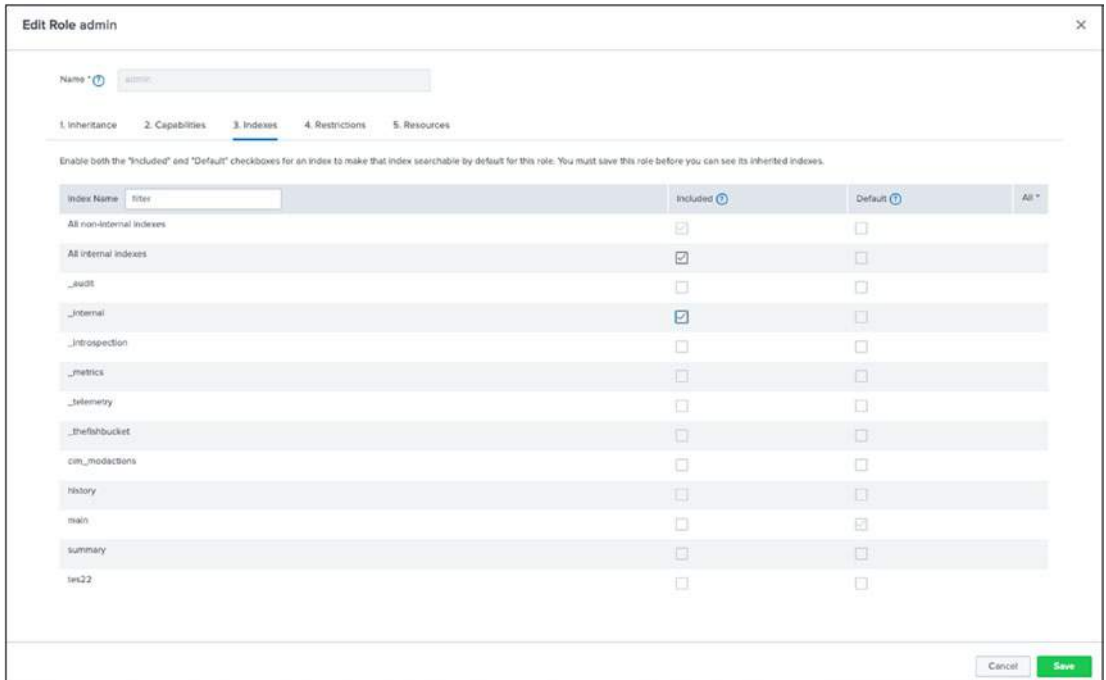
**Figure 8-8.** Defining Role Capabilities to Splunk Users

With edit roles and edit user capabilities, users can promote to a full admin role.

A new user in role index search does not need to specify an index name when searching. It is similar to the default index. In Role Index Access Options, if an index is not selected, users cannot access it.

1. Go to Settings and select Roles.
2. Go to Admin (select whichever user you want).
3. Go to Index Searched by default and assign or remove an index.

Figure 8-9 shows adding index access to Splunk users.



**Figure 8-9.** *Defining Index Access to Splunk Users*

You have now come to the end of the chapter. Commend yourself! You can test your knowledge by practicing with the MCQs at the end of the chapter.

## Summary

With the completion of Chapter 8, you have covered a big portion of the Splunk software basics. From tags and keywords to admins and licenses, you have learned most of Splunk's major concepts. This chapter dealt with Splunk licenses, license pooling, managing indexes, buckets, journal.gz, .tsidx, and Bloom filters, and user management. Therefore, you are now focusing on the executive elements and the administrative and management elements as well.

According to the Splunk Admin exam blueprint, you are familiar with 5% of Module 2 (licenses ) and 5% of Module 5 (user management and buckets).

The next chapter covers Splunk Forwarder and clustering.

## Multiple-Choice Questions

- A. Select the three main components of an index in Splunk.
  - 1. journal.gz
  - 2. .tsidx
  - 3. Bloom filter
  - 4. none of the above
  
- B. What is the maximum capacity of the Splunk Enterprise Trial License for indexing per day?
  - 1. 500 MB
  - 2. 700 MB
  - 3. 1 GB
  - 4. 100 MB
  
- C. Which are the preconfigured internal indexes? (Select all options that apply.)
  - 1. main
  - 2. \_internal
  - 3. \_audit
  - 4. \_introspection
  - 5. \_the fishbucket
  - 6. summary
  - 7. test
  - 8. none of the above

- D. You can create a Splunk index using the following options. (Select all options that apply.)
1. Splunk CLI
  2. indexes.conf
  3. Splunk Web
  4. inputs.conf
  5. transforms.conf
- E. In which phase of the index time process does license metering occur?
1. input phase
  2. parsing phase
  3. indexing phase
  4. licensing phase
- F. Local user accounts created in Splunk store passwords in which file?
1. \$ SPLUNK\_HCME/etc/users/authentication.conf
  2. \$ SPLUNK\_HOME/etc/passwd
  3. \$ SFLUNK\_KCME/etc/authentication
  4. \$ SFLUNK\_KOME/etc/passwd
- G. Metrics data is counted against the Enterprise license at a fixed 150 bytes per event.
1. true
  2. false
- H. A Splunk event in `_internal` index is counted against licensing.
1. true
  2. false

**Answers**

- A. 1, 2, 3
- B. 1
- C. 1, 2, 3, 4, 5, 6
- D. 1, 2, 3
- E. 3
- F. 2
- G. 1
- H. 2

**References**

- <https://docs.splunk.com/Documentation/Splunk/7.2.6/Indexer/Bloomfilters>
- <https://docs.splunk.com/Documentation/Splunk/latest/Indexer/HowSplunkstoresindexes>
- <https://docs.splunk.com/Documentation/Splunk/latest/Indexer/HowSplunkstoresindexes>
- <https://docs.splunk.com/Documentation/Splunk/7.2.6/Indexer/SmartStorecachemanager>
- <https://conf.splunk.com/files/2016/slides/behind-the-magnifying-glass-how-search-works.pdf>
- <https://docs.splunk.com/Splexicon:Bloomfilter>
- <https://docs.splunk.com/Splexicon:Tsidxfile>



## CHAPTER 9

# Machine Data Using Splunk Forwarder and Clustering

When handling a large amount of data from scores, data collection becomes necessary for the systematic procedure of the functions. Firstly, this chapter discusses the Splunk forwarder and its types. Secondly you will also see how to parse data from the Splunk forwarder into the Splunk indexer, demonstrating the power of machine data. Thirdly, you explore the process of clustering so that you can implement clusters into data collection. Finally you will see LDAP and SAML used for authentication and credential management in Splunk.

The following topics are covered in this chapter.

- Splunk universal forwarder
- Splunk light and heavy forwarder
- Forwarder management
- Splunk indexer clusters
- Splunk Lightweight Directory Access Protocol (LDAP)
- Splunk Security Assertion Markup Language (SAML)

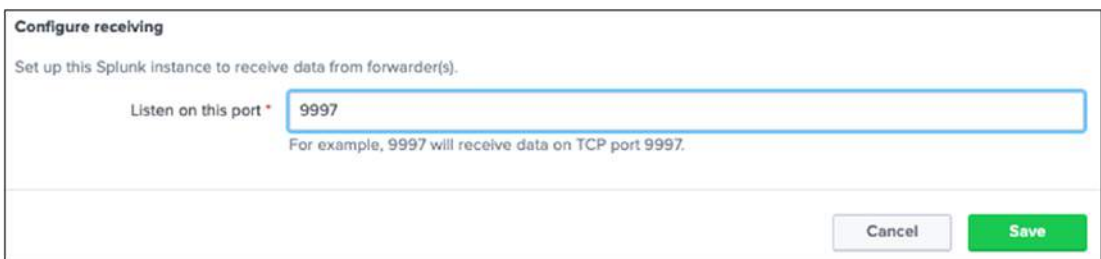
## Splunk Universal Forwarder

The Splunk universal forwarder collects data from the source and forwards it to the Splunk indexer using minimum hardware resources. It is lighter because it doesn't have a web interface, but the cost of installation needs to be considered. The universal forwarder includes only those components that are necessary to forward data to Enterprise instances. It is managed by editing configuration files, editing the forwarder manager, or through the Monitoring Console in the Splunk Enterprise instance.

## Configuring Splunk Indexer to Listen to Data for Universal Forwarder

The following explains the process of forwarding data into the Splunk indexer through Splunk Web.

1. Go to the Splunk indexer instance web interface.
2. Go to Settings and select Forwarding and Receiving.
3. Go to Configure Receiving and Add New.
4. Enter **9997** in **Listen on this port**. (Any port you wish, but if you allocate a different port, in a forwarder instance, you need to enter that port in port no block). Refer to Figure 9-1.



**Figure 9-1.** *Configure Listening On Indexer*

Any OS indexer platform is supported when sending logs from a Windows universal forwarder.

Do the following to enable port listening in the indexer using Splunk CLI.

1. In Linux, go to the Splunk indexer, and using the terminal, go to `$SPLUNK_HOME/bin/sudo./splunk` and enable `listen <port number>`.
2. In Windows, go to the Splunk indexer, and using Command Prompt, go to `%SPLUNK_HOME%/bin.\splunk` and enable `listen <port number>`.

Let's now look at the configuration used in the Windows Splunk forwarder.

## Configuring Windows Splunk Forwarder

There are two ways to configure the Windows Splunk universal forwarder.

- By using the command line
- By executing the .msi files

There are two ways to configure Windows Splunk forwarder. Windows can execute .msi files, or you can use the command line to install Splunk universal forwarder.

Download Splunk universal forwarder for Windows from [www.splunk.com/en\\_us/download/universal-forwarder.html](http://www.splunk.com/en_us/download/universal-forwarder.html).

## Splunk Universal Forwarder Using Windows

To configure Splunk universal forwarder using a command line, refer to the following steps.

1. Go to Command Prompt and go to `%SPLUNK_HOME%/bin`.
2. To start, stop, and restart Splunk, type `.\splunk start`, `.\splunk stop`, and `.\splunk restart`.
3. Forward data to the indexer by running the following command.

```
.\splunk add forward-server <IP or Hostname>:<Port>
```

4. Go to `SPLUNK_HOME/etc/system/local/outputs.conf` to check whether the universal forwarder is forwarding data or not. The following is the output.

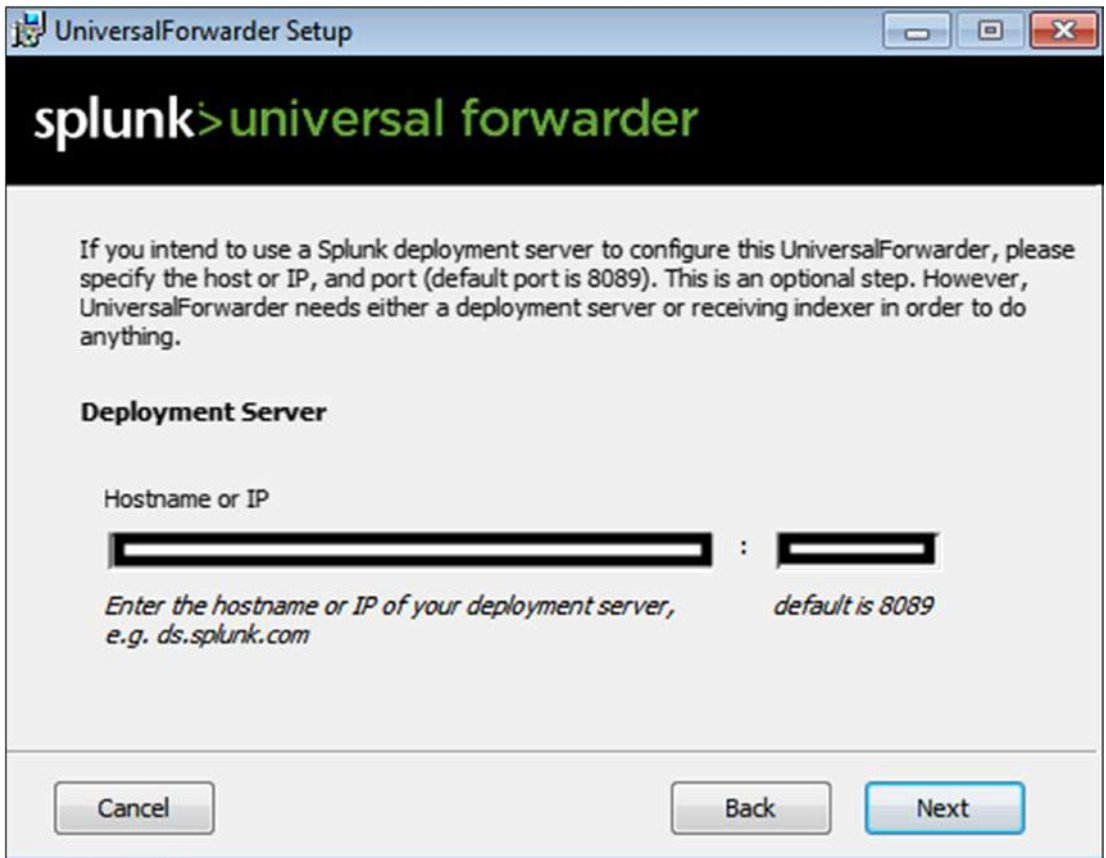
```
[tcpout]
defaultGroup = default-autolb-group

[tcpout:default-autolb-group]
disabled = false
server = 179.10.###.###:9997
```

## Splunk Universal Forwarder Using .msi

To configure Splunk universal forwarder using .msi, download the Splunk .msi file from [www.splunk.com/en\\_us/download/universal-forwarder.html](http://www.splunk.com/en_us/download/universal-forwarder.html). To configure universal forwarder, refer to the following instructions (see Figure 9-2).

1. Check the box to accept the license agreement, and then click Next.
2. Enter the username and the password.
3. Enter the deployment server hostname or IP and the port.



**Figure 9-2.** Configure Universal Forwarder To Indexer

Let's now look at configuring the Linux Splunk forwarder.

## Configuring Linux Splunk Forwarder

Linux Splunk forwarder can be downloaded through a command line widget, .rpm file, .tgz file, or .deb file. To install Splunk universal forwarder through an .rpm package file, .tgz package file, or a .deb package file, download from [www.splunk.com/en\\_us/download/universal-forwarder.html](http://www.splunk.com/en_us/download/universal-forwarder.html).

Let's now configure Splunk forwarder by using the terminal in Linux.

1. In the terminal, go to `$SPLUNK_HOME$bin`.
2. To start, stop, or restart Splunk, type `./splunk start`, `./splunk stop`, or `./splunk restart`.

3. Type **sudo./splunk start** to configure the Splunk instance.
4. Splunk asks if you agree. Type “**y**”. (y = yes and n = no).
5. Enter the administrator username as **admin** (or whatever you wish).
6. Enter a password.
7. Retype and confirm the password.
8. Type the following command to forward data to the indexer type.

```
./splunk add forward-server <IP or Hostname>:<Port>(9997)
```

9. Go to `$SPLUNK_HOME/etc/system/local/outputs.conf` to check whether the universal forwarder is forwarding data or not. You should see output similar to the following.

```
[tcpout]
defaultGroup = default-autolb-group

[tcpout:default-autolb-group]
disabled = false
server = 179.10.###.###:9997
```

Next, let’s discuss the two types of forwarders.

- Light forwarder
- Heavy forwarder

## Splunk’s Light and Heavy Forwarders

You have learned that a forwarder collects data from sources, forwards it to an indexer, and configures it using different methods. Now you learn about the two types of forwarders: heavy and light.

## Splunk Heavyweight Forwarder

A heavy forwarder has a smaller footprint than a Splunk indexer. It does almost all the jobs of the indexer except distributed searching. It parses data before forwarding and routes data based on criteria such as the source or type of event. A heavy forwarder also has the capability to index data locally and send it to a different indexer.

### Configuring Heavy Forwarder

To configure a heavy forwarder, observe the following steps.

1. Install Splunk Enterprise on the machine that you want to configure the heavy forwarder.
2. Click Settings and go to Forwarding and Receiving.
3. Select Configure Forwarding and click Add New.
4. Enter the deployment server hostname or IP and the port (Please enable listening port on indexer)

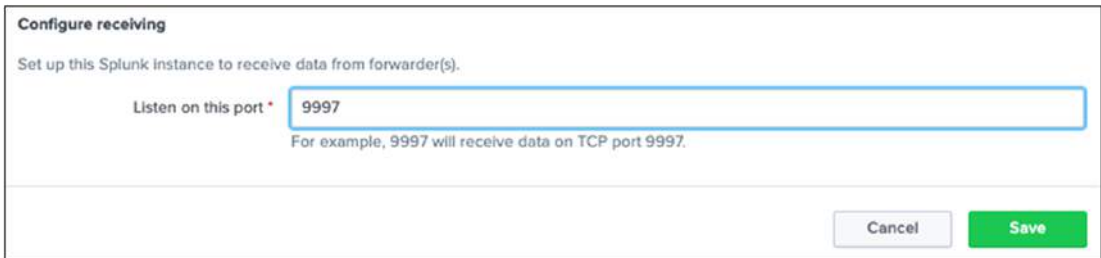
**Figure 9-3.** *Configure Splunk Heavyweight Forwarder To Indexer*

5. Restart the Splunk instance.

### Configuring Heavy Forwarder to Index and Forwarding Data from a Universal Forwarder

To configure a heavy forwarder to index and to forward the data, observe the following steps.

1. Go to the Splunk Enterprise instance web interface.
2. Click Settings and go to Forwarding and Receiving.
3. Go to Configure Receiving and click New.
4. In **Listen to this port**, type **9997**. (Use any port you wish but if you allocate a different port, you need to enter it in port no block in the forwarder instance.) Refer to Figure 9-4.



**Figure 9-4.** *Configure Receiving On Splunk Indexer*

The heavy forwarder is the type of Splunk forwarder that can parse data before forwarding.

## Splunk Light Forwarder

The light forwarder was deprecated from the Splunk Enterprise version 6.0, being replaced by the universal forwarder. So, configuration of the light forwarder is only possible on versions prior to Splunk Enterprise 6.0.

To forward and receive data in a Splunk light forwarder, configure both the receiver and the forwarder.

1. Install a full Splunk Enterprise instance.
2. Enable forwarding on the instance.

When you configure a Splunk instance as a light forwarder, select the forwarder license. It must be for a Splunk Enterprise version lower than version 6.0.

Let's now focus on forwarder management.



# Forwarder Management

This section discusses the management of the forwarder and its configuration.

Splunk forwarder management provides a centralized node to manage the entire Splunk domain. Forwarder Management role comprises of check the system's status, monitor deployment activity and configure and uninstall the apps in your environment. Forwarder Management Provides an interface to create server classes, mapping clients' deployment to the deployment of apps alongside. Its main objective is to check the system's status, monitor deployment activity, and configure and uninstall the apps in your environment. It allows you to modify the server classes and push an update. Simply put, forwarder management provides a centralized node to manage the entire Splunk domain.

There are three main tabs in forwarder management.

- **Apps:** This tab displays a list of the apps that are deployed and shows the app status. You can push app updates from inside forwarder management and edit the apps' properties.
- **Server classes:** This tab displays a list of classes that are deployed and shows the server-class status. You can create new server classes and edit the existing ones.
- **Clients:** This tab displays a list of the deployed clients and their status. You can restrict views of the app through this tab.

The Forwarder class in the forwarder management interface determines the apps that clients can install.

In the next section, you learn the various processes and platforms available to configure forwarder management.

## Configuring Forwarder Management

The following steps set up forwarder management in a Splunk environment.

The Forwarder Management screen resembles the one shown in Figure 9-5.



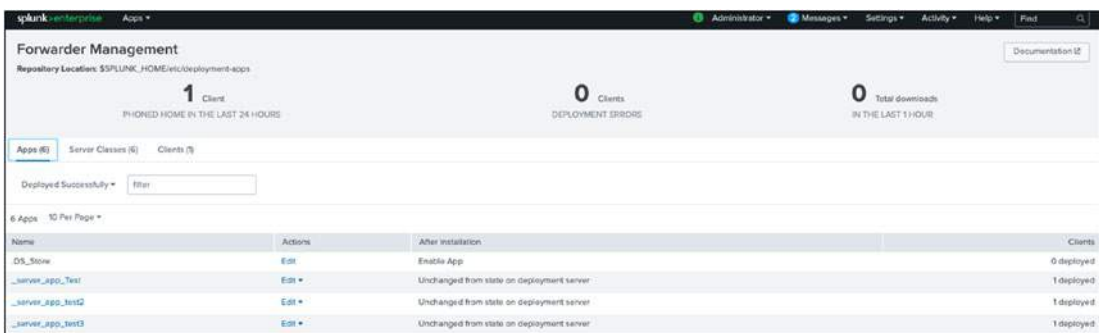
**Figure 9-5.** *Configuring Forwarder Management*

1. Go to `$SPLUNK_HOME/etc/system/local/serverclass.conf`. If this file does not exist, you need to create `serverclass.conf`.
2. Copy the following code into `serverclass.conf` (The code provided enables forwarder management in your splunk instance) and restart your instance.

```
[global]
[serverClass:AllApps]
[serverClass:AllApps:app:*]
```

3. Add the Test app to the deployment-server folder.
4. Go to Settings in Splunk Web and then go to Forwarder Management.

Figure 9-6 is a screenshot of the configured Forwarder Management page.



**Figure 9-6.** *Forwarder Management Page*

## Configuring the Forwarder Management Client

The forwarder management client can be a universal forwarder, indexer, or search head. All the components are potential clients for forwarder management except the deployment server (i.e., forwarder management).

To configure the client for forwarder management, refer to the following rules.

1. Using a Linux terminal, go to `$SPLUNK_HOME$/binsudo./splunk` and set `deploy-poll (IP or hostname):(mgmt port number)`
2. In Windows Command Prompt, go to `%SPLUNK_HOME%/binsudo.\splunk` and set `deploy-poll (IP or hostname):(mgmt port number)`

After setting `deploy-poll`, go to `$SPLUNK_HOME$/etc/system/local/deploymentclient.conf`. You find the `deployment client.conf` similar to the following.

```
[target-broker:deploymentServer]
targetUri= 179.12.XXX.XXX:8089
```

This sums up the discussion on the forwarder and its various elements. The next section discusses indexer clusters, which is another important aspect of data implementation.

## Splunk Indexer Clusters

A cluster in Splunk is the repetition factor number of copies of each bucket present on a peer node. Splunk indexer clusters consist of grouped indexers configured in a manner that enables the indexer to replicate data and maintain multiple copies of it. It supports automatic failover from one indexer to the next. As a result, if one indexer is down, the data coming into Splunk is indexed through another indexer and remains searchable.

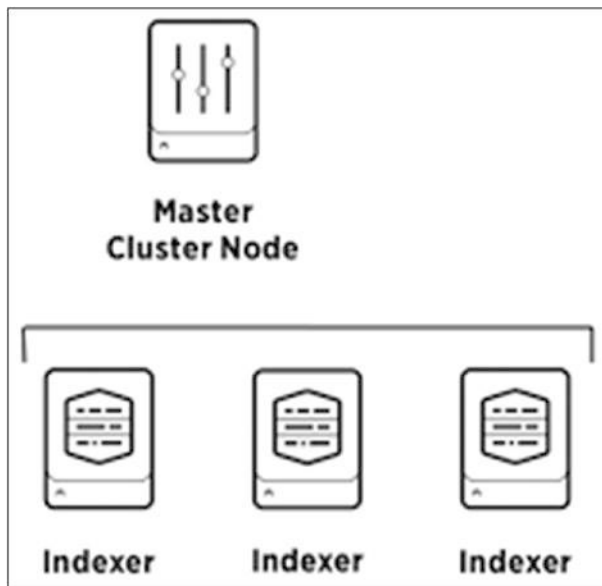
You can achieve the following points with index replication.

- Data availability
- Data fidelity
- Disaster tolerance
- Improved search performance

## Configuring Indexer Clusters

An indexer cluster is a group of Splunk Enterprise indexers that replicate data to create their backups. It also promotes high availability and disaster recovery through its data duplication and searching capabilities.

The basic indexer cluster architecture in Splunk includes two types of nodes (see Figure 9-7).



**Figure 9-7.** *Constituent Indexer Cluster Architecture*

- The **master node** manages the cluster. It keeps track of the replicating copies located at the peer nodes (slave nodes) and helps the search head find data. It also helps maintain automatic failover from one indexer to the next.
- The **peer node** receives and indexes incoming data from forwarders. A peer node can index its incoming data and simultaneously store copies of data from other nodes.

## Creating an Indexer Cluster Using Splunk Web

To create an indexer cluster using Splunk Web, refer to the following steps.

1. In Splunk Web, go to Settings.
2. Select Indexer Clustering and enable it.

This is where you distinguish how master nodes and peer nodes are created.

To create a master node, enter the following.

1. Enter the replication factor. (A higher replication factor protects against loss of data if peer nodes fail.)
2. Enter the search factor (A higher search factor speeds up the time to recover lost data at the cost of disk space.)
3. Enter the security key. (This key authenticates communication between the master and the peers.)
4. Name your cluster. (This step is optional.).

Figure 9-8 shows an example of configuring a master node.

**Master Node Configuration** [X]

**Replication Factor**   
The number of copies of raw data that you want the cluster to maintain. A higher replication factor protects against loss of data if peer nodes fail.

**Search Factor**   
The number of searchable copies of data the cluster maintains. A higher search factor speeds up the time to recover lost data at the cost of disk space. Must be less than or equal to Replication Factor.

**Security Key**   
This key authenticates communication between the master and the peers and search heads.

**Cluster Label**   
Name your cluster using this field. This label is also used to identify this cluster in the Monitoring Console.

**Figure 9-8.** *Configuring Master Node: Splunk Web*

Once the required details are filled in, click Enable Master Node to create the node. To configure a peer node, enter the following information.

1. Enter the master URI (Ip:mgmt\_port/host\_name:mgmt\_port).
2. Enter the peer replication port. (The port peer nodes use to stream data to each other.)

3. Enter the security key. (This key authenticates communication between the master and the peers.)
4. Click Enable Peer Node.

Figure 9-9 shows an example of configuring a peer node.

**Figure 9-9.** *Configuring Peer Node: Splunk Web*

## Creating an Indexer Cluster Using a Splunk .conf File

To configure a Splunk indexer cluster from the .conf file for a master node, the indexer creates or edits `indexer.conf` in `$Splunk_Home$/etc/system/local`.

The syntax for configuring a master node using a .conf file is shown in the following code block.

```
[clustering]
mode = master
replication_factor = <number>
```

```
search_factor = <number>  
pass4SymmKey = <String>  
cluster_label = <String>
```

- Replication\_factor protects against loss of data if peer nodes fail.
- search\_factor speeds up the time to recover lost data at the cost of disk space.
- pass4SymmKey authenticates communication between the master and the peers.
- cluster\_label is the name of the cluster.
- To configure a peer node, the indexer creates or edits indexer.conf located in \$Splunk\_Home\$/etc/system/local.

The syntax for configuring a peer node using a .conf file is shown in the following code block.

```
[replication_port://<port>]  
  
[clustering]  
master_uri = https://<ip>:<mgmt_port>,<host_name>:<mgmt_port>  
mode = slave  
pass4SymmKey = <string>
```

- master\_uri is the address of the master node.
- pass4SymmKey authenticates communication between the master and the peers.

## Creating an Indexer Cluster Using Splunk CLI

To configure an indexer cluster using Splunk CLI for a master node, refer to the following steps.

1. Using Windows CMD or a Linux terminal, go to %SPLUNK\_HOME%/bin.
2. To start, stop and restart Splunk, type **.\splunk start**, **.\splunk stop**, and **.\splunk restart**.



The syntax for configuring a master node using Splunk CLI is shown in the following code block.

```
splunk edit cluster-config -mode master -replication_factor <number>
-ssearch_factor <number> -secret <string> -cluster_label <string>

splunk restart
```

The following describes the steps for a peer node.

1. Using Windows CMD or a Linux terminal, go to %SPLUNK\_HOME%/bin.
2. To start, stop, and restart Splunk, type **.\splunk start**, **.\splunk stop**, and **.\splunk restart**.

The syntax for configuring a peer node using Splunk CLI is shown in the following code block.

```
splunk edit cluster-config -mode slave -master_uri https://<ip>:<mgmt_
port>,<host_name>:<mgmt_port> -replication_port <port> -secret <string>
-cluster_label <string>

splunk restart
```

Let's now move on to Splunk LDAP, which supervises authentication in the Splunk domain.

## Splunk Lightweight Directory Access Protocol (LDAP)

You have learned about the Splunk forwarder, forwarder management, and indexer clusters. Together they are responsible for the collection, allocation, and gathering of data from sources. This section discusses data authentication and credential management.

Splunk LDAP plays a critical role in the authentication of the Splunk Web. It maintains user credentials like user IDs and passwords and information like hostname, bind DN, and user base DN. When you configure LDAP in Splunk Web, it handles authentication.

Each time the user logs in, Splunk rechecks the LDAP server.

## Creating an LDAP Strategy

To create an LDAP strategy, follows these steps.

1. Click Settings in the Splunk home page and go to Authentication Methods.
2. Click Configure Splunk to use LDAP and go to New LDAP. In the New LDAP, enter the following details.
  - a. LDAP strategy name
  - b. The hostname of your LDAP server (ensure that the Splunk server can identify the hostname)
  - c. The port that Splunk Enterprise uses to connect to the LDAP server defaults to port 636 (by default, LDAP servers listen on TCP port 389 and LDAPS (LDAP with SSL))
3. Check if SSL is enabled. This is an optional step.
4. Provide the bind DN. (It defines the domain name of an administrator.)
5. Enter and confirm the bind DN password for binding user
6. Specify the user base DN (Splunk Enterprise uses this attribute to locate user information)

The following is other information that may need to be addressed.

- Userbase filter (recommended to return only applicable users; for example, (department=IT, Management, Accounts, etc..))
- User name
- Real name attribute
- Mail address
- Group mapping attribute (a user attribute that group entries use to define their members)
- Group base DN (the location of the user groups in LDAP)

- Static group search filter for the object class you want to filter your static groups on
- Group name attribute (a group entry attribute whose value stores the group name)
- Static member attribute (a attributes group attribute whose values are the group's members)

For nested groups, check Nested groups. Enter the Dynamic group search filter to retrieve dynamic groups; if it exists then, enter the Dynamic member attribute

In Advanced Settings, you can limit the search request time, search request size limit, and networking socket timeout.

7. Click Save and exit.

Figures 9-10 and 9-11 show the process of LDAP strategy creation.

LDAP strategy name \*   
Enter a unique name for this strategy.

**LDAP connection settings**

Host \*   
Your Splunk server must be able to resolve this host.

Port   
The LDAP server port defaults to 389 if you are not using SSL, or 636 if SSL is enabled.

SSL enabled  
You must also have SSL enabled on your LDAP server.

Bind DN   
This is the distinguished name used to bind to the LDAP server. This is typically the DN of an administrator with access to all LDAP users you wish to add to Splunk. However, you can leave this blank if anonymous bind is sufficient.

Bind DN Password   
Enter the password for your Bind DN user.

Confirm password

**User settings**

User base DN \*   
The location of your LDAP users, specified by the DN of your user subtree. If necessary, you can specify several DNs separated by semicolons.

User base filter   
The LDAP search filter used to filter users. Highly recommended if you have a large amount of user entries under your user base DN. For example, '(department=IT)'

User name attribute \*   
The user attribute that contains the username. Note that this attribute's value should be case insensitive. Set to 'uid' for most configurations. In Active Directory (AD), this should be set to 'sAMAccountName'.

Real name attribute \*   
The user attribute that contains a human readable name. This is typically 'cn' (common name) or 'displayName'.

Email attribute   
The user attribute that contains the user's email address. This is typically 'mail'.

Group mapping attribute   
The user attribute that group entries use to define their members. If your LDAP groups use distinguished names for membership you can leave this field blank.

Figure 9-10. Creating LDAP Strategy

**Group settings**

Group base DN \*   
 The location of your LDAP groups, specified by the DN of your group subtree. If necessary, you can specify several DNs separated by semicolons.

Static group search filter   
 The LDAP search filter used to retrieve static groups. Highly recommended if you have a large amount of group entries under your group base DN. For example, '(department=IT)'

Group name attribute \*   
 The group attribute that contains the group name. A typical value for this is 'cn'.

Static member attribute \*   
 The group attribute whose values are the group's members. Typical values are 'member' or 'memberUid'. Groups list user members with values of groupMappingAttribute, as specified above.

Nested groups  
 Controls whether Splunk will expand nested groups using the 'memberof' extension. Only check this if you have nested groups and the 'memberof' extension on your LDAP server.

**Dynamic group settings**

Dynamic member attribute   
 The dynamic group attribute that contains the LDAP URL used to find members. This setting is required to configure dynamic groups. A typical value is 'memberURL'.

Dynamic group search filter   
 The LDAP search filter used to retrieve dynamic groups (optional). For example, '(objectclass=groupOfURLs)'

Advanced settings

Cancel Save

**Figure 9-11.** *Creating LDAP Strategy*

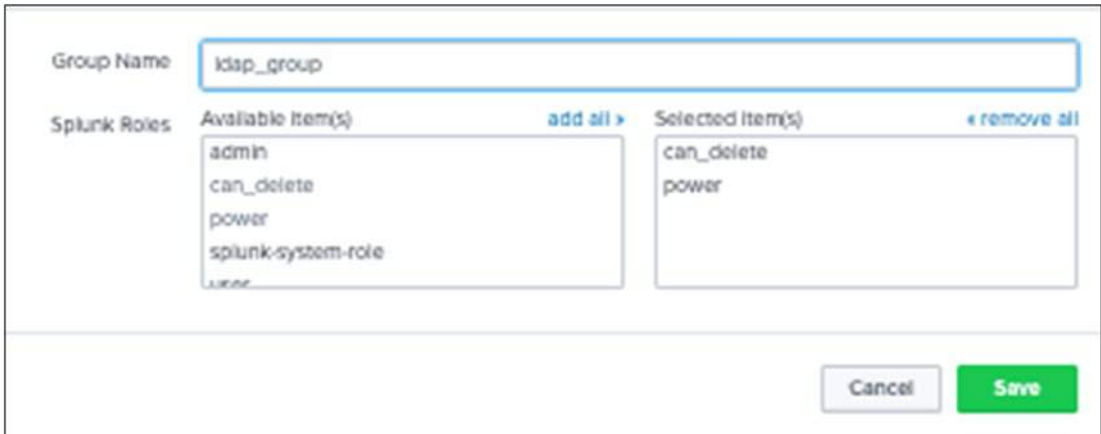
## Mapping LDAP Group to Splunk Roles

Splunk Enterprise authenticates and maps the various roles via the LDAP server.

To map the LDAP group to Splunk roles, follow these instructions.

1. Select LDAP.
2. Configure Splunk to LDAP and map groups.
3. Select Map Groups in the Actions column for a specific strategy.
4. Click Group Name.
5. Map a role to a group and click the arrow to a role in the Available Roles list. This moves the group into the Selected Roles list, which helps multiple roles to be mapped in the group (see Figure 9-12).

6. Click Save.



**Figure 9-12.** Mapping LDAP Strategy: Splunk Roles

That summarizes the Splunk LDAP analysis, its creation, mapping, and its importance in the Splunk domain. Let’s move to Splunk SAML.

## Splunk Security Assertion Markup Language (SAML)

Splunk SAML (Security Assertion Markup Language) is an identity provider (IdP) that maintains users’ credentials and handles authentication. When you configure a Splunk instance to use the SAML authentication system, you authorize groups onto your IdP and enable their login by mapping them to Splunk user roles. SAML, however, doesn’t support authentication.

Splunk SAML and LDAP can also be configured using authentication.conf, located in \$SPLUNK\_HOME/etc/system/local.

## Configuring Splunk SAML

To configure Splunk SAML, refer to the following steps.

1. Click Settings and go to Authentication Methods.
2. Click Configure Splunk to use SAML and create a new SAML.

3. Copy the metadata file directly.
4. In General Settings, provide the following data.
  - Single sign-on URL (automatically populated when you enter the copy metadata file)
  - IdPs Certificate Path (It can be a directory or a file).
  - Entity ID (It is the entity's identity)
1. Select **Sign auth request**.
2. Select **Sign SAML response**.
3. If you use PingIdentity as your IdP, specify the attribute query URL, the sign attribute query request, and the sign attribute query response.
4. In Advance Settings, configure information like attribute alias role, attribute alias role name, and attribute alias mail.
5. Click Save.

Figure 9-13 illustrates the process of Splunk SAML configuration.

**SAML Configuration**

Configure SAML for Splunk. [Learn More](#)

Download the SP Metadata from Splunk and add it to your SAML environment to connect to Splunk.

SP Metadata File

Import Identity Provider (IDP) metadata by browsing to an XML file, or copy and paste the information into the Metadata Contents text box.

Metadata XML File

Metadata Contents

**General Settings**

Single Sign On (SSO) URL

Single Log Out (SLO) URL

IDP certificate path   
Leave blank if you store IDP certificates under \$SPLUNK\_HOME/etc/auth/IdpCerts

IDP certificate chains

Replicate Certificates

Issuer Id

Entity ID

Sign AuthRequest

Verify SAML response

► Attribute Query Requests

► Authentication Extensions

► Alias

**Advanced Settings**

Name Id Format

Fully qualified domain name or IP of the load balancer

Redirect port - load balancer port

Redirect to URL after logout

SSO Binding

SLO Binding

**Figure 9-13.** Configuring SAML

Let's look at mapping SAML.



## Map SAML to User Roles

To map the SAML to Splunk roles, refer to the following instructions.

1. Go to Access Controls and select Authentication Method.
2. Select SAML and go to Configure Splunk to SAML and map groups.
3. On the SAML Groups page, click New Group, or click Edit to modify the existing group.
4. Provide a name for the group.
5. Determine the roles you want to assign to this group by moving the desired roles from the left column to the right column.
6. Click Save.

Figure 9-14 is an illustration of the SAML mapping method.

The screenshot shows a 'Create New SAML Group' dialog box. It has a title bar with a close button (X). The main content area is divided into sections. The first section is 'Group Name' with a text input field containing 'test'. The second section is 'Splunk Roles', which is split into two columns. The left column is labeled 'Available Item(s)' and contains a list of roles: 'admin', 'can\_delete', 'power', 'splunk-system-role', and 'user'. The right column is labeled 'Selected Item(s)' and contains a list of roles: 'can\_delete' and 'power'. Between the two columns are two buttons: 'add all >' and '< remove all'. At the bottom right of the dialog, there are two buttons: 'Cancel' and 'Save'.

**Figure 9-14.** Mapping SAML: Splunk Roles

This sums up the working of the Splunk SAML.

You have come to an end of this chapter. You should congratulate yourself on the successful learning of Splunk data management. This chapter proves important when the need to assort or simplify data arises and takes you a step further in Splunk administration.

## Summary

This chapter covered Splunk universal forwarder, the heavy and light forwarders, Splunk indexer clusters and their configuration, Splunk LDAP, and Splunk SAML. These topics administer the gathering, assembling, and authentication of the data in a Splunk environment. Data collection often includes remote sources and large-scale processing. Instance forwarders and indexer clusters ensure that all the data is assigned to the right buckets and sorted into the correct nodes. Splunk LDAP and SAML ensure data input is authenticated and the users' credentials are authorized. You learned implementation, configuration, and role mapping in step-by-step discussions.

The next chapter discusses how to fetch data using monitor and network inputs, extracting data using scripted and Windows input, pulling data using agentless input, and data sanitizing using fine-tuned input.

Practice what you learned in this chapter and test your knowledge with the following questions.

## Multiple-Choice Questions

- A. Which forwarder type can parse data before forwarding?
  1. universal forwarder
  2. heaviest forwarder
  3. light forwarder
  4. heavy forwarder
  
- B. Which Splunk indexer operating system platform is supported when sending logs from a Windows universal forwarder?
  1. any OS platform
  2. Linux platform only
  3. MAC OS platform only
  4. none of the above

- C. Which Splunk component requires a forwarder license?
1. search head
  2. heavy forwarder
  3. heaviest forwarder
  4. universal forwarder
- D. Which of the following authentication types requires scripting in Splunk?
1. ADFS
  2. LDAP
  3. SAML
  4. RADIUS
- E. How often does Splunk recheck the LDAP server?
1. every 5 minutes
  2. each time a user logs in
  3. each time Splunk is restarted
  4. varies based on the LDAP\_refresh setting
- F. Which of the following statements describe deployment management?
1. requires an Enterprise license
  2. is responsible for sending apps to forwarders.
  3. once used, is the only way to manage forwarders
  4. automatically restarts the host OS running the forwarder
- G. Which authentication methods are natively supported within Splunk Enterprise? (select all that apply)
1. ADFS
  2. LDAP
  3. SAML
  4. RADIUS

- H. How can you configure the Splunk index in Splunk? (Select all that apply.)
1. forwarder management
  2. Splunk CLI
  3. indexes.conf
  4. Splunk Web

**Answers**

- A. 4
- B. 1
- C. 2
- D. 4
- E. 2
- F. 2
- G. 2, 3
- H. 2, 3, 4

## References

- <https://docs.splunk.com/Documentation/Splunk/7.2.6/Security/SetuptoolsauthenticationwithLDAP>
- <https://docs.splunk.com/Documentation/Splunk/7.2.6/Security/HowSAMLSSOworks>
- <https://docs.splunk.com/Documentation/Splunk/7.2.6/Security/ConfigureSSOPing>
- <https://docs.splunk.com/Documentation/Forwarder/7.2.6/Forwarder/Abouttheuniversalforwarder>
- <https://docs.splunk.com/Splexicon:Heavyforwarder>
- <https://docs.splunk.com/Documentation/Splunk/7.2.6/Forwarding/Deployaforwarder>

## CHAPTER 10

# Advanced Data Input in Splunk

This chapter advances the discussion of forwarders and input. You explore getting data from monitor input, network input, scripted input, and agentless input.

The following topics are covered in this chapter.

- Additional forwarding options
- Fetching data using monitor input
- Scripted input
- Network input
- Pulling data using agentless input

In the first section, you analyze additional forwarding options. Through these options, you can compress a data feed before sending data to a Splunk indexer and acknowledge the indexed data to prevent data loss. These additional forwarding options secure a data feed so that you can maintain data integrity in Splunk.

- **Compress the data feed:** Compressing a data feed in Splunk can be done by editing `outputs.conf` in the forwarding environment and `inputs.conf` on the receiving side. If you don't compress the data feed, the forwarding environment directly sends the raw events. However, this option compresses the raw events before transferring.
- **Indexer acknowledgment:** Indexer acknowledgment in Splunk is done by editing `outputs.conf` in forwarding environmental `inputs.conf` on the receiving side. Using indexer acknowledgment, the forwarder resends any data not acknowledged as received by the indexer.

- **Securing the feed:** Securing the feed (SSL) in Splunk is done through `outputs.conf` and `sever.conf`. To add SSL on the receiving side, you can edit `inputs.conf`. Once you have secured your feed using the default SSL certificate or with your SSL, you can maintain data integrity. Additionally, there is no need to compress the data because it is done automatically.
- **Forwarding environment queue size:** Queue size in Splunk is modified through the `.conf` maximum queue size, which is 500 KB. It is the maximum amount of data the forwarder has when it cannot reach out to the server.

Let's now discuss their commands and syntax.

## Compress the Data Feed

Splunk compresses raw data from the forwarding environment and sends it to the receiving side. On the receiving side, you need to decompress the data feed to get raw data and increase CPU usage. `compressed=true` lets the forwarder environment compress the data before it sends data to receivers.

To compress the data feed in the forwarder environment, edit `outputs.conf` as shown in the following code block.

```
[tcpout:splunk_indexer]
server=<ip>:<receiving_port>/<hostname>:<receiving_port>
compressed=true
```

To compress the data feed on the receiving side, edit `inputs.conf` using the following code block command.

```
[splunktcp:<receiving_port>]
compressed=true
```

With this, you have completed learning the additional forwarding options and the compression of raw data syntax. Next, let's discuss indexer acknowledgement.

## Indexer Acknowledgment

Indexer acknowledgment helps prevent data loss when the forwarding environment is forwarding data to the receiving side. The forwarding environment resends the data if the Splunk indexer does not acknowledge it. When you use indexer acknowledgement in the forwarder, the wait queue should be modified to 3x so that the requirement for larger space is met.

To allow the indexer to acknowledge the forwarder environment, edit `outputs.conf` as shown in the following code block.

```
[tcpout:splunk_indexer]
server=<ip>:<receiving_port>/<hostname>:<receiving_port>
useACK=true
```

To allow the indexer to acknowledge the forwarder environment, edit the `inputs.conf`, as shown in the following code block.

```
[splunktcp:<receiving_port>]
useACK=true
```

---

When you use `Ack=true`, the wait queue that manages indexer acknowledgment has a default maximum size of 21 MB in Splunk, so the output queue in the indexer should be 7 MB because the wait queue is 3x the output queue.

---

## Securing the Feed

Securing the feed (SSL certificate) is important so that no one can easily send data to the indexer or manipulate the events. To add SSL on the receiving side, you need to edit `inputs.conf`, and to add it on the sender side, you need to edit `outputs.conf` and `server.conf`. Securing the feed using SSL increases CPU usage and automatically compresses the data feed.

To secure the feed in the forwarder environment, edit `outputs.conf` as shown in the following code block.

```
[tcpout:splunk_indexer]
server=<ip>:<receiving_port>/<host_name>:<receiving_port>
sslPassword=<password>
clientCert=<path for server.pem file>
sslVerifyServerCert=false
```

To secure the feed in the forwarder environment in Linux, edit the `server.conf` as shown in the following code block.

```
[sslConfig]
sslRootCAPath=<path for cacert.pem file>
```

To secure the feed in the forwarder environment in Windows, edit `server.conf` as follows. (Only for Windows)

```
caCertFile = <location of cacert.pem file>
caPath = <path of cacert.pem file>
```

To secure the feed in the receiving environment, edit `inputs.conf` as follows.

```
[splunktcp:<receiving_port>]
[ssl]
sslPassword=<password>
serverCert=<path for server.pem file>
requireClientCert=false
```

To secure the feed in the receiving environment, edit `server.conf` (not in Windows) as follows.

```
[sslConfig]
sslRootCAPath=<path for cacert.pem file>
```

Let's discuss queue sizes in the following section.

## Queue Size

In a Splunk forwarding environment, the queue size manages indexer acknowledgement. You cannot configure a wait queue size directly; instead, you only configure the output queue size in `outputs.conf`. By default, the maximum queue size is 500 KB if `useACK=false`; if `useACK=true`, the maximum output queue size is 7 MB, and the wait queue is 21 MB. The recommended `maxQueueSize` attribute settings are `auto` because it optimizes queue size based on whether indexer acknowledgment is active or not.



To set the queue size to auto in the forwarder environment, edit `outputs.conf` as shown in the following code block.

```
[tcpout:splunk_indexer]
server=<ip>:<receiving_port>/<hostname>:<receiving_port>
maxQueueSize=auto
```

## Monitor Input

A monitor input is the act of watching a file, directory, script output, or network ports for new data. It can also add data from files and directories. There are two types of wildcards that you can add in `inputs.conf` to monitor files/directories. They are explained in Table 10-1.

**Table 10-1.** *Wildcards for Monitor Inputs*

Wildcard	Description
...	The ellipsis wildcard recurses through directories and subdirectories until a match.
*	The asterisk wildcard matches files/directories anything in the specific directory path.

## Monitor Files

A monitor input defines a specific file as a data source; it performs the following tasks on a data source.

- The current data of the file is inserted.
- The file is continuously monitored for new content.
- Splunk tracks the file status, even after restarting a Splunk instance. It automatically points to the location from where the Splunk instance stopped tracking the file.

A file monitor in Splunk supports the following data types.

- Plain text data files
- Structured text files

- Multiline logs
- The ability to read compressed files

## Monitor Directories

A monitor input defines a specific directory as a data source. Monitor input performs the following tasks in a data source.

- Splunk recursively travels through a directory structure
- All types of files in the directory are included for monitoring

In a monitor input, you can use blacklists and whitelists to monitor files and directories.

- **Whitelist:** A filtering rule to include one or more members from a set. You use whitelist rules to tell a forwarder which files to consume when monitoring directories. You can define a whitelist in `inputs.conf`.
- **Blacklist:** A filtering rule to exclude one or more members from a set. You use blacklist rules to tell a forwarder which files not to consume when monitoring directories. You can define a blacklist in `inputs.conf`.

---

Monitor files support CSV, XML, and JSON formats.

---

## Monitor Files and Directory Using Splunk Web

To monitor files and directories using Splunk Web, refer to the following instructions. In this section, you monitor the `KDC-setup.log` file in which clients want files continuously monitored.

1. Go to Settings and go to Add Data.
2. Go to monitor and select Files & Directories.
3. Click Browse. Navigate to files. Select the file. Refer to Figure 10-1 for reference. (I selected the `KDC-Setup.log` file to be monitored.)

**Figure 10-1.** *Monitor:KDC-Setup.log*

4. Click Continuously Monitor or Index Once, depending on your requirements. I selected the Continuous Monitor option.
5. Click Next to go to the Set Source Type page (select the source type based on your data). Refer to Figure 10-2.

	Time	Event
1	10/31/18 9:09:48.000 AM	Wed Oct 31 09:09:48 IST 2018 lkdc restore trigger re-setup of LKDC on next boot
2	10/31/18 9:11:31.000 AM	Wed Oct 31 09:11:31 IST 2018 creating system keychain entries ...System identity already exists for domain com.apple.systemdefault. Done. ...System identity already exists for domain com.apple.kerberos.kdc. Done. /System/Library/PrivateFrameworks/MeIndal.framework/Helpers/Adc already in acl Show all 6 lines

**Figure 10-2.** *Sourcetype catalina:KDC-setup.log*

6. Click Next to go to the Input Settings page. Select App: Test.
7. In the Index Data Type field, verify that the default events index is selected, and click Save.

8. Back on the Input Settings page, the test should be displayed for the index value.
9. Click Submit.

## Monitor File and Directory Using inputs.conf

You can directly monitor files or directories in Splunk using inputs.conf.

You can directly monitor files by writing inputs.conf. Refer to the following steps to configure inputs.conf to monitor files in Splunk.

1. Using the terminal, go to \$SPLUNK\_HOME\$/etc/system/local/inputs.conf. (If it doesn't exist, create a new file named inputs.conf).
2. Edit inputs.conf. Add a stanza in inputs.conf.
3. Select Add Source (after monitor:// in the stanza header). It is the file path.
4. Select All Attributes (source type, host, index, blacklist, whitelist are optional but can be added to inputs.conf).

Refer to Table 10-2 for a description of the attributes.

**Table 10-2.** Monitor File:inputs.conf

Attribute	Value
disabled=<true false>	When disabled= input script won't run and when disabled= input script invokes
Whitelist	A filtering rule to include one or more members from a set. You use whitelist rules to tell a forwarder which files to consume while monitoring directories
Blacklist	A filtering rule to exclude one or more members from a set. You use Blacklist rules to tell a forwarder which files not to consume while monitoring directories
host=<hostname>	Splunk Enterprise uses the hostname during parsing and indexing. A hostname is used at searching.

*(continued)*

**Table 10-2.** (continued)

Attribute	Value
index=<Index name>	Splunk Enterprise sets an index to save an event.
Sourcetype=<sourcetype name>	Source type in Splunk Enterprise is used for formatting data during parsing and indexing.
source=<sourcename>	Source name in Splunk Enterprise is an input file path where data originates.

As an example, if clients want to continuously monitor the KDC-setup.log file, you can directly write a stanza using inputs.conf. (Refer to the following solution.)

1. Go to \$SPLUNK\_HOME/etc/apps/Test/local/inputs.conf. Create a stanza as follows.

```
[monitor:///Library/Logs/KDC-setup.log]
disabled=false
sourcetype=Test18
```

2. Save the changes.
3. Restart the Splunk instance.

Next, let's discuss the scripted input.

## Scripted Input

Splunk scripted input can accept data from scripts. You can use scripted input to get data to Splunk Enterprise from a third-party application, application program interface (API), remote data interface, and so on. Scripted input onboards data to Splunk Enterprise. Scripted input can be added to Splunk instances through Splunk Web or by editing inputs.conf. When you configure a Splunk environment to run a scripted input, it clears any environment variables that can affect the script's operation.

Splunk execute scripts from \$SPLUNK\_HOME/etc/apps/<app\_name>/bin,\$SPLUNK\_HOME/bin/scripts or \$SPLUNK\_HOME/etc/system/bin.

## Scripted Input Using Splunk Web

In scripted input, you write a script to get the total size of RAM, the size of 1 block of RAM, total used blocks, and the interval every 60 seconds. After every 60 seconds, the local instance using scripted input sends data to the Splunk instance. The following is reference code in Python.

```
import sys
print("Total size",sys.getsizeof({}))
print("size of 1 block",sys.getsizeof([]) )
print("used block",sys.getsizeof(set()) )
```

1. Go to Settings and go to Add Data.
2. Go to Monitor and select Scripts.
3. Select the script path where your script is placed. (Normally place your script in the app's bin folder.
4. Select the script from the drop-down.
5. In the interval, select seconds or cron schedule, whichever method you want.
6. Provide parameters based on whether the interval you select is seconds or cron schedule. Refer to Figure 10-3 for reference.

**Add Data** ● ○ ○ ○ < Back Next >

Select Source    Input Settings    Review    Done

**Files & Directories**  
Upload a file, index a local file, or monitor an entire directory.

**HTTP Event Collector**  
Configure tokens that clients can use to send data over HTTP or HTTPS.

**TCP / UDP**  
Configure the Splunk platform to listen on a network port.

**Scripts**  
Get data from any API, service, or database with a script. >

Configure this instance to execute a script or command and to capture its output as event data. Scripted inputs are useful when the data that you want to index is not available in a file to monitor. [Learn More](#)

Script Path:

Script Name:

Command:

Interval Input:

Interval:

Source name override:

**FAQ**

- > What kind of scripts can I run?
- > How do I control when scripts run?
- > How do I write my own Cron Schedule?

**Figure 10-3.** Adding Script:1.py

7. Click Next.
8. In the Input Settings page.
  - a. Select the source type based on your data. You can also create a new source type. (The source type is one of the default fields that the Splunk platform assigns to all incoming data. It tells the Splunk platform what kind of data you have so that it can intelligently format the data during indexing.)
  - b. In-App, select the test app.
  - c. In Source Type, select Test18.
  - d. In Index, select the index named *Test*.
  - e. Click Next.
  - f. Click Review.
  - g. Click Finish. (Refer to Figure 10-4.)





**Table 10-3.** *Scripted Input:inputs.conf*

Attribute	Value
disabled=<true false>	When disabled= input script won't run and when disabled= input script invokes.
interval=<number> cron schedule	You can specify the time interval. Set interval=<value>. After that interval, the input script runs again. Set interval=<cron>. When you specify a cron schedule, the script does not execute on startup, but rather at the times that the cron schedule defines.
host=<hostname>	Splunk Enterprise uses the hostname during parsing and indexing. A hostname is used at the time of searching.
index=<Index name>	Splunk Enterprise sets an index where it should save an event.
Sourcetype=<sourcetype name>	Source type in Splunk Enterprise is used for formatting of data during parsing and indexing of data.
source=<sourcename>	Source name in Splunk Enterprise is an input file path from where data originates.

In scripted input, you write a script to get the total size of RAM, the size of one block of RAM, total used blocks, and the interval as every 60 seconds. After every 60 seconds, the local instance using scripted input sends data to the Splunk instance. You write a script stanza in `inputs.conf` to monitor scripted input.

1. Copy 1.py code in the Test application bin folder.
2. Go to `$SPLUNK_HOME/etc/apps/Test/local/inputs.conf`. Create a stanza as follows.

```
[script://$SPLUNK_HOME/etc/apps/Test/bin/1.py]
disabled=false
sourcetype=Test18
interval=60
index=Test
```

3. Save the changes Restart the Splunk instance.

After completing the Scripted input, you now move on to the next section - the Network Inputs.

## Network Input

Splunk Enterprise can accept input on any TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) port. Splunk can consume any data on these ports. syslog is a good example of network-based data. TCP is the recommended protocol for sending data in Splunk, but UDP is used for monitoring data. Network input in Splunk can monitor a network port on a local machine, or you can receive network data from another machine using forward.

With a TCP input, you need to specify whether the port should accept connections from all hosts or specific hosts. The host value can be IP, DNS, or custom (user-defined label). To set up network input in Splunk, the network protocol and port number need to be specified.

## Add Network Input Using Splunk Web and Deploy It to the Forwarder

The following explains how to add a network input in Splunk.

1. In the Splunk indexer, click Settings, go to Add Data, and click Forward.
2. In Select Forwarders, configure the form and provide the information. Create a server class, provide the hostname, and provide an appropriate server class.
3. In Select Source, click TCP/UDP and configure. Provide the port on which you want to listen to data. Provide a source name, and you can configured to accept data from a particular forwarder. Refer to Figure 10-5 for help.

**Add Data**

Select Forwarders   Select Source   Input Settings   Review   Done

< Back   Next >

**Files & Directories**  
Upload a file, index a local file, or monitor an entire directory.

**TCP / UDP**  
Configure the Splunk platform to listen on a network port.

**Scripts**  
Get data from any API, service, or database with a script.

Configure selected Splunk Universal Forwarders to listen on any TCP or UDP port to capture data sent over the network from services such as syslog. [Learn More](#)

TCP   UDP

Port ? 9000  
Example: 514

Source name override ? test12  
host:port

Only accept connection from ? optional  
example: 10.1.2.3, !badhost.splunk.com, \*splunk.com

**FAQ**

- > How should I configure the Splunk platform for syslog traffic?
- > What's the difference between receiving data over TCP versus UDP?
- > Can I collect syslog data from Windows systems?
- > What is a source type?

**Figure 10-5.** TCP Network Input Listening Port:9000

4. For advance input settings, configure the form and provide source type and index name. Refer to Figure 10-6 for help.

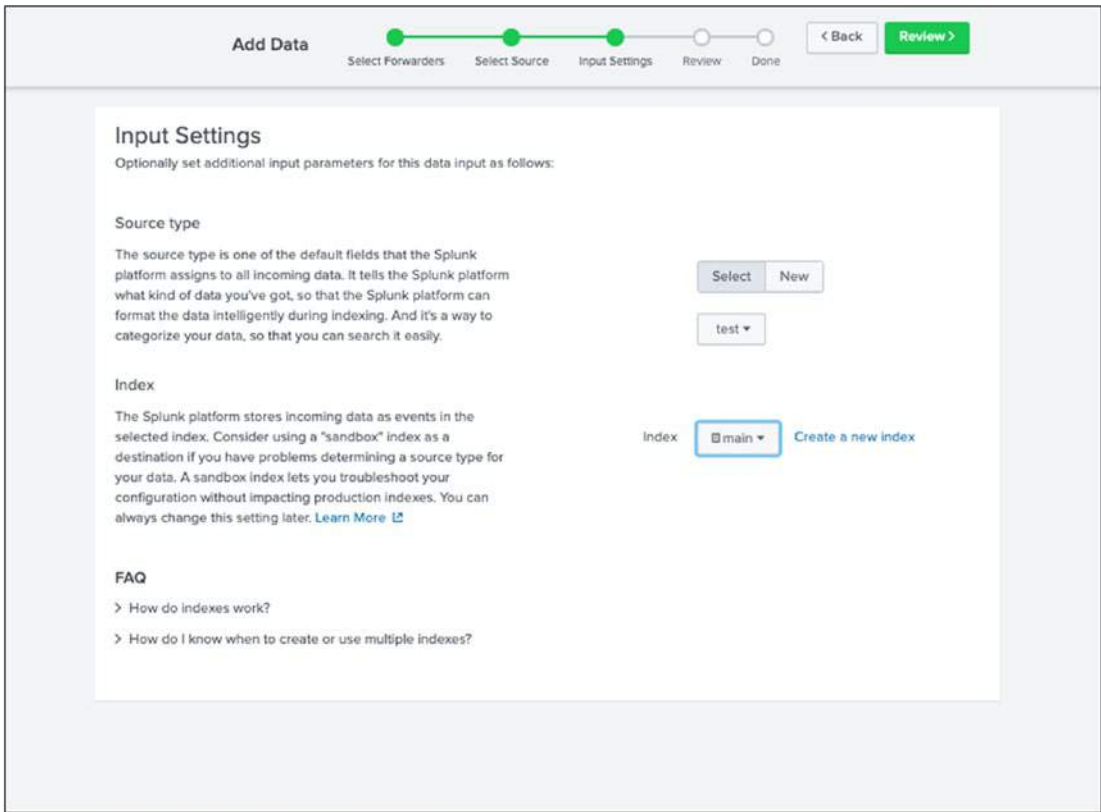


Figure 10-6. Network Input: Advance Input Settings

## Modify Network Input Using .conf Files

You can modify network input using .conf files. You can change the index value, source value, and so forth using .conf files.

## Configure TCP Network Input Using .conf File

To configure TCP network input, go to `SPLUNK_HOME/etc/system/local/inputs.conf`. In the `inputs.conf` stanza, `[tcp://<host or ip>:<port>]`, there are various parameters that you can configure (see Table 10-4).

**Table 10-4.** *TCP Network Input:inputs.conf*

attribute	Value
host=<hostname>	Splunk Enterprise uses the hostname during parsing and indexing. A hostname is used at searching.
index=<Index name>	Splunk Enterprise sets an index where it should save an event.
Sourcetype=<sourcetype name>	The source type in Splunk Enterprise is used for formatting of data during parsing and indexing of data.
source=<sourcenam>	The source name in Splunk Enterprise is an input file path from where data originates.
queue = parsingQueue   indexQueue	Queue specifies where the input processor should deposit event that it reads.
connection_host=ipldns	Ip in Splunk Enterprise sets the host to the IP address of the Splunk server. DNS in Splunk Enterprise sets the host to the DNS entry of the Splunk server.

If clients want to monitor all system logs listening on Transmission Control Protocol port 514, you can write a TCP stanza in `inputs.conf`. Refer to the following example.

1. Go to `$$SPLUNK_HOME/etc/apps/Test/local/inputs.conf`. Create a stanza as follows.

```
[tcp://514}
connection_host=dns
sourcetype=Test18
index=Test
```

2. Save the changes.
3. Restart the Splunk instance.

## Configure Network UDP Input Using .conf File

To configure a UDP network input, go to `SPLUNK_HOME/etc/system/local/inputs.conf`. In the `inputs.conf` stanza, `[udp://<host or ip>:<port>]`, there are various parameters that you can configure (see Table 10-5).

**Table 10-5.** *UDP Network Input:inputs.conf*

attributeAttribute	Value
host=<hostname>	Splunk Enterprise uses the hostname during parsing and indexing. A hostname is used at searching.
index=<Index name>	Splunk Enterprise sets an index where it should save an event.
Sourcetype=<sourcetype name>	The source type in Splunk Enterprise is used for formatting of data during parsing and indexing of data.
source=<sourcename>	The source name in Splunk Enterprise is an input file path from where data originates.
queue = parsingQueue   indexQueue	Queue specifies where the input processor should deposit event that it reads.
_rcvbuf=<value>	_rcvbuf in Splunk Enterprise sets the receive buffer for UDP.
no_priority_stripping = true   false	no_priority_stripping sets how Splunk Enterprise handles receiving Syslog data.
no_appending_timestamp = true   false	no_appending_timestamp Sets how Splunk Enterprise applies timestamps and hosts to events.

If clients want to monitor all Application Performance Monitoring (APM) logs listening on User Datagram Protocol port 9514, you can write a UDP stanza in `inputs.conf`. Refer to the following example.

1. Go to `$$SPLUNK_HOME/etc/apps/Test/local/inputs.conf`. Create stanza as follows.

```
[tcp://514}
connection_host=ip
sourcetype=Test18
index=Test
```

2. Save the changes.
3. Restart the Splunk instance.

Let's next discuss pulling data using agentless input.

## Pulling Data Using Agentless Input

An HTTP Event Collector (HEC) is a token-based HTTP input that is secure and scalable. The HEC allows you to send data and application events to a Splunk Enterprise instance using HTTP and Secure HTTP (HTTPS) protocols. HEC uses a token-based authentication model.

An agent-based HTTP input is more secure and scalable and doesn't need to use forwarders.

## HTTP Input Using Splunk Web

In this section, you create an HTTP event, configure the HEC, and modify a stanza using a configuration file. Then, you parse data from a local instance to a Splunk Enterprise instance using HEC tokens.

To get HTTP input using Splunk Web, refer to the following steps.

1. Go to Settings ► Data Inputs. Go to HTTP Event Collector and click New Token.
2. Enter the name **Test**.
3. Enter the source name and description (optional). If you want to enable indexer acknowledgment for this token, don't click the Enable indexer acknowledgment checkbox (in this exercise).
4. Click Next (see Figure 10-7.)

Configure a new token for receiving data over HTTP. [Learn More](#)

Name

Source name override <sup>?</sup>

Description <sup>?</sup>

Output Group (optional)

Enable indexer acknowledgement

**Figure 10-7.** *Configure HTTP Input:Test*

5. (Optional) Select the source type and the index(test) for HEC events.
6. Click Review (see Figure 10-8).



## Input Settings

Optionally set additional input parameters for this data input as follows:

### Source type

The source type is one of the default fields that the Splunk platform assigns to all incoming data. It tells the Splunk platform what kind of data you've got, so that the Splunk platform can format the data intelligently during indexing. And it's a way to categorize your data, so that you can search it easily.

Automatic Select New

### App context

Application contexts are folders within a Splunk platform instance that contain configurations for a specific use case or domain of data. App contexts improve manageability of input and source type definitions. The Splunk platform loads all app contexts based on precedence rules. [Learn More](#)

App Context Test (Test) ▾

### Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. [Learn More](#)

Select Allowed Indexes Available Item(s) add all > Selected item(s) < remove all

history  
 main  
 summary  
 test

test

Select indexes that clients will be able to select from.

Default Index  Create a new index

### FAQ

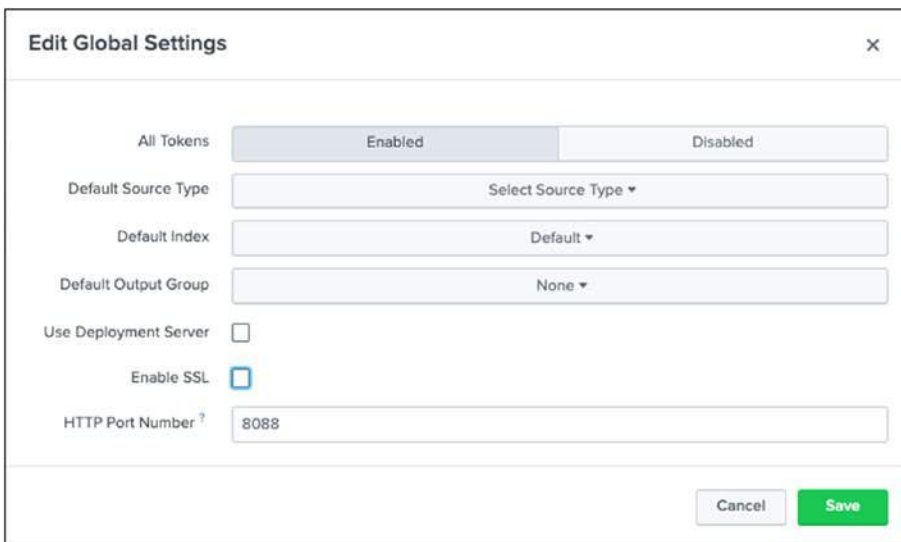
- > How do indexes work?
- > How do I know when to create or use multiple indexes?

**Figure 10-8.** HTTP Events:Advanced Input Settings

## Configure HTTP Event Collector in Splunk

Before you can use an HTTP Event Collector in Splunk, you need to enable it in global settings in Splunk Web. You also need to configure your firewall and network settings. To configure the HEC to receive events in Splunk Web, refer to the following instructions

1. Click Settings and go to Data Inputs.
2. Click HTTP Event Collector.
3. Click Global Settings
4. In All tokens, toggle it to Enable.
5. Select Source Type, Index, and Output group (Optional).
6. To use a deployment server to handle configurations for HEC tokens, click the use Deployment Server checkbox.
7. To have HEC listen and communicate over HTTPS rather than HTTP, click the Enable SSL Checkbox. (For this exercise, don't enable it.)
8. Enter the HTTP port number (whichever port number you want) to make it listen to HTTP events.
9. Click Save (see Figure 10-9).



**Figure 10-9.** HTTP Event collector:Enabled

## Configure HTTP Input Using .conf File

To configure HTTP input using a .conf file, go to `SPLUNK_HOME/etc/apps/splunk_httpinput/local/inputs.conf`. In `inputs.conf`, create a stanza `[http]`. In that stanza, there are various parameters (see Table 10-6).

**Table 10-6.** *Configure HTTP input:inputs.conf*

Attribute	Value
<code>disabled=&lt;0 1&gt;</code>	When <code>disabled=&lt;0&gt;</code> HTTP input is enabled and when <code>disabled=&lt;1&gt;</code> HTTP input is disabled.
<code>host=&lt;hostname&gt;</code>	Splunk Enterprise uses the hostname during parsing and indexing. A hostname is used at searching.
<code>index=&lt;Index name&gt;</code>	Splunk Enterprise sets an index where it should save an event.
<code>Sourcetype=&lt;sourcetype name&gt;</code>	The source type in Splunk Enterprise is used for formatting of data during parsing and indexing of data.
<code>source=&lt;sourcenam&gt;</code>	The source name in Splunk Enterprise is an input file path from where data originates.
<code>enableSSL=0 1</code>	When <code>enableSSL=&lt;0&gt;</code> SSL is off and when <code>enable SSL =&lt;1&gt;</code> SSL is enabled.

To update HTTP input using a direct configuration file in this exercise, enable SSL for HTTP input.

1. Go to `$SPLUNK_HOME/etc/apps/splunk_httpinput/local/inputs.conf`. Create a stanza as follows.

```
[http]
disabled=0
enableSSL=1
```

2. Save the changes.
3. Restart the Splunk instance.

## Configure HTTP Event Collector in Splunk Using .conf File

To configure an HTTP Event Collector to receive events in Splunk Web, you need to enable HEC (HTTP Event Collector) through global settings.

4. Go to `$SPLUNK_HOME/etc/apps/$APP_NAME/local/inputs.conf`.
5. In `inputs.conf`, create a stanza, `[http://<HEC Name>]`. In that stanza, there are various parameters (see Table 10-7).

**Table 10-7.** *Configure HTTP Event Collector:inputs.conf*

Attribute	Value
<code>disabled=&lt;0 1&gt;</code>	When <code>disabled=&lt;0&gt;</code> HTTP input is enabled, and when <code>disabled=&lt;1&gt;</code> HTTP input is disabled.
<code>token=&lt;value&gt;</code>	Token in HTTP helps in authentication, and the value of the token is Unique. Easy to identify. Token value is called a <i>globally unique identifier</i> .
<code>host=&lt;hostname&gt;</code>	Splunk Enterprise uses the hostname during parsing and indexing. A hostname is used at searching.
<code>index=&lt;Index name&gt;</code>	Splunk Enterprise sets an index where it should save an event.
<code>Sourcetype=&lt;sourcetype name&gt;</code>	The source type in Splunk Enterprise is used for formatting of data during parsing and indexing of data.
<code>source=&lt;sourcename&gt;</code>	The source name in Splunk Enterprise is an input file path from where data originates.
<code>useACK=0 1</code>	When <code>useACK=&lt;0&gt;</code> indexer acknowledgement is disabled, and when <code>useAck =&lt;1&gt;</code> indexer acknowledgement is enabled.

Next, let's modify the HTTP Event collector in Splunk using the `inputs.conf` file where you enable acknowledgement.

1. Go to `/$SPLUNK_HOME/etc/apps/Test/local/inputs.conf`. Modify the stanza as follows.

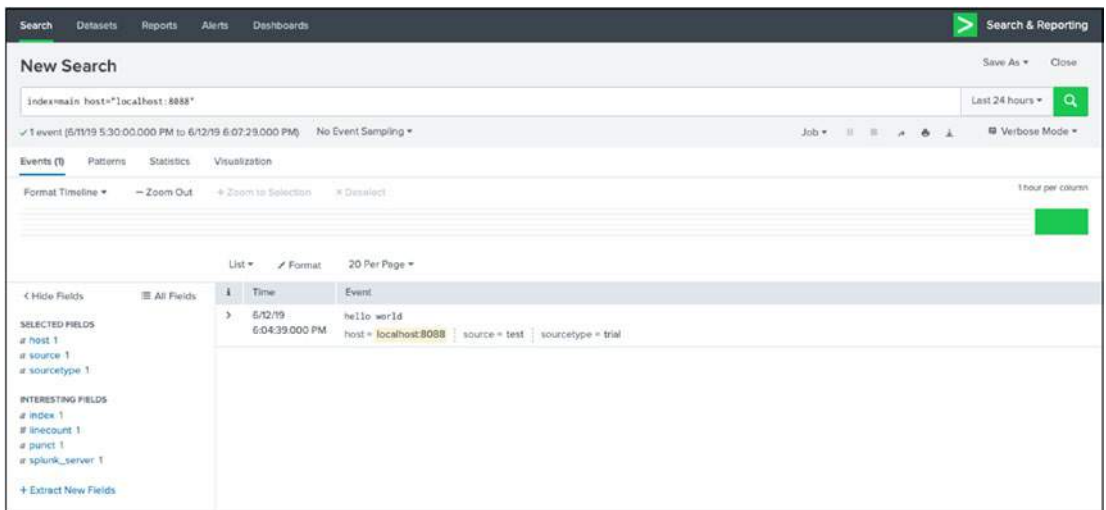
```
[http://Test]
disabled=0
indexes=Test
Token=XXXXXXXX
useACK=1
index=Test
```

2. Save the changes.
3. Restart the Splunk instance.

## Parse Data in Splunk Using HTTP Event Collector

Now, you can parse data to a Splunk instance using an HTTP Event Collector. You can refer to the following code block and Figure 10-10 to better understand how to parse data in a Splunk instance using HEC.

```
curl -k https://localhost:8088/services/collector -H "Authorization: Splunk 876ba4bc-ff77-46f9-b4a4-2b6dd4bcfd14" -d '{"sourcetype": "trial" , "event": "hello world"}'
```



**Figure 10-10.** HTTP Event Collector event:Hello World

You have reached the end of this chapter. I would like to congratulate you on learning about additional forwarding options, monitor input, scripted input, and network inputs, which are responsible for the bulk of data in the Splunk environment.

## Summary

You learned how to get data from various sources, including monitor input, network input, scripted input, and agentless input. You can now enter data in a more defined manner while ensuring that the events are not invaded or altered. You can also enable the security of the feed and control and modify the queue size. You now know that acknowledge indexers are one of the most important parts of Splunk software. You can check your understanding using the following questions.

## Multiple-Choice Questions

- A. Which of the following enables compression for universal forwarders in `outputs.conf`?
  1. `[udpout:mysplunk_indexer1] compression=true`
  2. `[tcpout]defaultGroup=my_indexers compressed=true`
  3. `/opt/splunkforwarder/bin/splunk enable compression`
  4. `[tcpout:my_indexers] server=mysplunk_indexer1:9997,mysplunk_indexer2:9997 decompression=false`
  
- B. A universal forwarder has which capabilities when sending data? (Select all that apply.)
  1. sending alerts
  2. compressing data
  3. obfuscating/hiding data
  4. indexer acknowledgment

- C. Which of the following statements apply to directory inputs?  
(Select all that apply.)
1. All discovered text files are consumed.
  2. Compressed files are ignored by default.
  3. Splunk recursively traverses through the directory structure.
  4. When adding new log files to a monitored directory, the forwarder must be restarted to take them into account.
- D. How does the Monitoring Console monitor forwarders?
1. by pulling internal logs from forwarders
  2. by using the forwarder monitoring add-on
  3. with internal logs forwarded by forwarders
  4. with internal logs forwarded by the deployment server
- E. When configuring monitor input with whitelists or blacklists, what is the supported method of filtering the lists?
1. slash notation
  2. regular expression
  3. irregular expression
  4. wildcard-only expression
- F. To set up a network input in Splunk, what needs to be specified?
1. file path
  2. username and password
  3. network protocol and port number
  4. network protocol and MAC address

- G. Which option accurately describes the purpose of the HTTP Event Collector (HEC)?
1. a token-based HTTP input that is secure and scalable and that requires the use of forwarders
  2. a token-based HTTP input that is secure and scalable and that does not require the use of forwarders
  3. an agent-based HTTP input that is secure and scalable and that does not require the use of forwarders
  4. a token-based HTTP input that is insecure and non-scalable and that does not require the use of forwarders
- H. What is the difference between the two wildcards ... and \* for the monitor stanza in inputs.conf?
1. ... is not supported in monitor stanzas
  2. no difference—they are interchangeable and match anything beyond directory boundaries
  3. \* matches anything in that specific directory path segment, whereas ... recurses through subdirectories as well
  4. ... matches anything in that specific directory path segment, whereas \* recurses through subdirectories as well

**Answers**

- A. 2
- B. 2, 4
- C. 1, 3, 4
- D. 3
- E. 2
- F. 3
- G. 2
- H. 3



## References

- <https://docs.splunk.com/Splexicon:Scriptedinput>
- <https://docs.splunk.com/Documentation/Splunk/7.3.0/Data/Monitornetworkports>
- <https://docs.splunk.com/Documentation/Splunk/7.3.0/AdvancedDev/ScriptSetup>
- <https://docs.splunk.com/Documentation/Splunk/7.3.0/Data/MonitorWindowsnetworkinformation>
- <https://docs.splunk.com/Documentation/Splunk/7.3.0/Data/UsetheHTTPEventCollector>

## CHAPTER 11

# Splunk's Advanced .conf File and Diag

This chapter discusses the various .conf files responsible for storing Splunk configuration information and apps. You learn about fine-tuning the input. You study the process of anonymizing sensitive personal data and learn about debugging configuration files. You are introduced to the diag, which collects basic information on the environment, the instance, and the operating system.

In a nutshell, the following are topics covered in this chapter.

- Understanding Splunk .conf files
- Setting fine-tuning input (custom source type)
- Anonymizing data
- Understanding merging logic in Splunk
- Debugging a configuration file
- Creating a diag

## Understanding Splunk .conf files

The configuration file is called .conf file in Splunk. A Splunk configuration file contains Splunk Enterprise configuration information.

Default configuration files are saved in `$SPLUNK_HOME/etc/system/default`. (It is advised to edit local files in Splunk and avoid editing default files because they are overridden whenever Splunk is updated). To edit local files, go to `$SPLUNK_HOME/etc/system/local`. Stanzas in the configuration files include system settings, authentication and authorization information, index-related settings, deployment, cluster configuration, knowledge objects, and searches in Splunk.

The following files are discussed in this section.

- props.conf
- indexes.conf
- transforms.conf
- inputs.conf
- outputs.conf
- deploymentclient.conf

## props.conf

The props.conf file is present on the heavy forwarder, search head, and indexer. In this file, you can apply parsing rules on data. The props.conf file is mainly used for the following.

- Configuring line breaking for events
- Character encoding
- Processing of binary files
- Timestamp recognition and converting time formats
- Event segmentation

The props.conf file is located in `$SPLUNK_HOME/etc/system/local/props.conf`. Table 11-1 describes the file's attributes and their respective values.

**Table 11-1.** *Configuring props.conf*

Attribute	Value
SHOULD_LINE_MERGE=false true	SHOULD_LINE_MERGE specifies whether to combine several lines of data into a single multiline event.
TIME_PREFIX=<regular expression>	The RegEx of an event checks for the timestamp in an event.
TIME_FORMAT=<strftime-style format>	TIME_FORMAT specifies the “strftime” format string and extracts the date from the event.
MAX_TIMESTAMP_LOOKAHEAD=<Integer>	MAX_TIMESTAMP_LOOKAHEAD specifies the number of characters that Splunk should check in a timestamp.
LINE_BREAKER=<regular expression>	LINE_BREAKER identifies the start of the next event.
BREAK_ONLY_BEFORE=<regular expression>	BREAK_ONLY_BEFORE is a method for defining the start of the next event.
MAX_EVENTS=<Integer>	MAX_EVENTS specifies the maximum number of lines for an event.
ANNOTATE_PUNCT=false true	ANNOTATE_PUNCT enables or disables punctuation in an event for searching.
KV_MODE = <none KV pairs>	KV_MODE =none is used when you don't have any kV pairs.
SEDCMD-<class> = y/<string1>/<string2>/	SEDCMD-<class> hides data and can be replaced with other data.

Table 11-1 consists of a few of the important instructions that are used in props.conf. For more information, refer to <https://docs.splunk.com/Documentation/Splunk/7.3.1/Admin/propsconf>.

## indexes.conf

The `indexes.conf` file exists on the Splunk indexer mainly to configure indexes and manage index policies, such as data expiration and data thresholds. In the `indexer.conf` file, you can apply rules for creating indexes in the `Splunk.indexes.conf` file, which is primarily used for configuring indexes and their properties. The `indexes.conf` file is located in `$SPLUNK_HOME/etc/system/local/indexes.conf`.

Table 11-2 describes how to configure the `indexes.conf` file.

**Table 11-2.** *Configuring indexes.conf*

Attribute	Value
<code>homePath=&lt;Path&gt;</code>	<code>homePath</code> contains the location of hot and warm buckets.
<code>coldPath=&lt;Path&gt;</code>	<code>coldPath</code> contains the location of cold buckets.
<code>thawedPath=&lt;Path&gt;</code>	<code>thawedPath</code> contains the location of thawed database.
<code>repFactor=&lt;int &gt;</code>	Determines whether the indexer needs to be replicated or not.
<code>maxHotBuckets=&lt;int&gt;</code>	The maximum number of hot buckets that can exist in the index.
<code>maxDataSize=&lt;int&gt;</code>	It specifies the maximum data that can be saved in an index.
<code>maxWarmDBCount=&lt;int&gt;</code>	It specifies the maximum number of warm buckets.
<code>maxTotalDataSizeMB=&lt;int&gt;</code>	It specifies the maximum size of a particular index.
<code>frozenTimePeriodInSecs=&lt;int&gt;</code>	The number of seconds after which the index rolls to frozen.
<code>coldToFrozenDir=&lt;Path&gt;</code>	It specifies a path for the frozen archive.

Table 11-2 lists some of the important instructions in `indexes.conf`. For more information, refer to <https://docs.splunk.com/Documentation/Splunk/7.3.1/Admin/Indexesconf>

## transforms.conf

The `transforms.conf` file is present on the light forwarder, heavy forwarder, search head, and indexer. In this file, you can apply the rules for parsing data and applying regular expression rules to transform source type, anonymize certain types of sensitive incoming data, and create advanced search field extraction. The `transforms.conf` file is located in `$SPLUNK_HOME/etc/system/local/transforms.conf`.

Table 11-3 describes how to configure the transforms.conf file.

**Table 11-3.** *Configuring transforms.conf*

Attribute	Value
SOURCE_KEY=<_raw>	SOURCE_KEY indicates which data stream to use as the source for pattern matching.
REGEX=<regular expression>	REGEX identifies events from SOURCE_KEY.
DEST_KEY=<Metadata:Sourcetype>	DEST_KEY indicates where to write the processed data
FORMAT=<Sourcetype::<sourcetype name>>	FORMAT controls how REGEX writes the DEST_KEY.

Table 11-3 lists a few of the important instructions in transforms.conf. For more information, go to <https://docs.splunk.com/Documentation/Splunk/8.0.0/Admin/Transformsconf>.

## inputs.conf

The inputs.conf file exists on the universal forwarder, heavy forwarder, search head, and indexer. However, there are various attributes in the inputs.conf file that need to be handled before inputting data. The inputs.conf file is located in \$SPLUNK\_HOME/etc/system/local/inputs.conf.

Table 11-4 illustrates the configuration of inputs.conf file.

**Table 11-4.** *Configuring inputs.conf*

Attribute	Value
host=<string>	It sets the host value to the static value for the input stanza.
index=<string>	It specifies the index to store events that comes through the input stanza.
source=<string>	It sets the source field for the events from the input field.
sourcetype=<string>	It sets the source type field for the events from these input field.
host_regex=<regular expression>	Th regular expression extracts the host from the path to the host file.
whitelist=<regular expression>	The files from this input are monitored if the path matches the regular expression.
blacklist=<regular expression>	The files from this input are not monitored if the path matches the regular expression.
outputgroup=<string>	The name of the group where event collector forwards data.
enableSSL=0 1	It decides whether the HTTP event collector group should use SSL or not.
maxSockets=<integer>	It decides the count of the HTTP connections that the HTTP event collector input accepts simultaneously.
queue=parsingQueue indexQueue	It tells the index where it should submit the incoming events for a particular input stanza.

Table 11-4 lists a few of the important instructions. For more information on inputs.conf, refer to <https://docs.splunk.com/Documentation/Splunk/7.3.1/Admin/Inputsconf#>.

## outputs.conf

The outputs.conf file is present on the universal forwarder, the heavy forwarder, the search head, and the indexer. In this file, you can apply rules for sending data out to Splunk instances. However, various attributes need to be handled before the forwarder sends data to the receiving Splunk instances. The outputs.conf file is located in \$SPLUNK\_HOME/etc/system/local/outputs.conf.

Table 11-5 illustrates the configuration of the `outputs.conf` file.

**Table 11-5.** *Configuring `outputs.conf`*

Attribute	Value
<code>heartbeatFrequency=&lt;integer&gt;</code>	Heartbeat frequency notifies the receiving server by sending a heartbeat package specified in the integer block.
<code>maxQueueSize=&lt;integer&gt;</code>	It indicates the RAM size of all items in the queue.
<code>autoLBFrequency=&lt;integer&gt;</code>	It is used for load balancing, specifying the time in the integer block to change the server.
<code>autoLBVolume=&lt;integer&gt;</code>	The volume of data in bytes to send to an indexer before the new one is selected from available indexers.
<code>maxEventSize=&lt;integer&gt;</code>	The maximum size of an event that Splunk can transmit.
<code>server=&lt;ip&gt;:&lt;port&gt;, &lt;servername&gt;:&lt;port&gt;</code>	It specifies the target indexer for transmitting the data.
<code>timestampformat=&lt;format&gt;</code>	It specifies the prepended formatted timestamp format for events.
<code>master_uri=&lt;ip&gt;:&lt;port&gt;, &lt;servername&gt;:&lt;port&gt;</code>	The address of the master Uri of the cluster.
<code>type=tcpludp</code>	It specifies whether you want to use a TCP or UDP protocol.
<code>compressed=truelfalse</code>	It specifies whether data needs to be compressed or not.
<code>_TCP_ROUTING=&lt;target_group&gt;</code>	This specifies the target group to forward data.
<code>disabled=truelfalse</code>	It determines whether it needs to disable events transmitting or not.

Table 11-5 lists a few of the important instructions in `outputs.conf`. For more information, refer to <https://docs.splunk.com/Documentation/Splunk/7.3.1/Admin/outputsconf>.



## deploymentclient.conf

The `deploymentclient.conf` file resides on the universal forwarder, the heavy forwarder, the search head, and the indexer. This file contains descriptions of the settings that you can use to customize a deployment client's behavior. There are various attributes in the `deploymentclient.conf` files that are useful for setting the client-server communication. The `deploymentclient.conf` file is located in `$SPLUNK_HOME/etc/system/local/deploymentclient.conf`.

Table 11-6 shows the various attributes and their values in the `deploymentclient.conf` file.

**Table 11-6.** *Configuring deploymentclient.conf*

Attribute	Value
<code>targerUri=&lt;uri&gt;</code>	Specifies the address of the deployment server.
<code>connect_timeout=&lt;integer&gt;</code>	The maximum time in seconds that a deployment client can take to connect to the deployment server.
<code>send_timeout=&lt;integer&gt;</code>	The maximum time in seconds that a deployment client can take to send or write data to the deployment server.
<code>recv_timeout=&lt;integer&gt;</code>	The maximum time in seconds that a deployment client can take to read or receive data to the deployment server.

Table 11-6 lists a few of the important instructions in `deploymentclient.conf`. For more information, refer to <https://docs.splunk.com/Documentation/Splunk/7.3.1/Admin/deploymentclientconf>.

You have completed the discussion on the six configuration files in Splunk. In the next section, you learn about the fine-tuning inputs that create customized source types for input data.

## Setting Fine-Tuning Input

The Fine-tuning inputs in Splunk are responsible for creating custom source types for data, which are scripted input, unusual log files, custom data types, and others. You can also create your source types by modifying `props.conf` or in Splunk Web so that Splunk can understand all that data.

## Custom Source Types Using Splunk Web

In this section, you create a custom source type for test.txt so that Splunk can recognize events, extract the timestamp from events, and understand where a new event is starting. To do this for your data using Splunk Web, refer to the following instructions.

1. Go to Settings and head over to Source types.
2. Click New Source types.
3. Enter the name of source type as **test**.
4. In Destination, app, select the test app.
5. In Category, select Custom Data.
6. Splunk has predefined indexed extraction based on the type of data you can select it. For test.txt, select none. (For a better understanding, refer to Figure 11-1.)

The screenshot shows the 'Create Source Type' dialog box in Splunk Web. The dialog has a title bar with 'Create Source Type' and a close button (X). Below the title bar are several input fields and dropdown menus:

- Name:** A text input field containing 'test'.
- Description:** A text input field containing 'optional'.
- Destination app:** A dropdown menu showing 'test'.
- Category:** A dropdown menu showing 'Custom'.
- Indexed Extractions<sup>?</sup>:** A dropdown menu showing 'none'.

Below these fields are three tabs: 'Event Breaks', 'Timestamp', and 'Advanced'. The 'Event Breaks' tab is selected and underlined. Under this tab, there is a section titled 'Define event boundaries for incoming data.' with an 'Event-breaking Policy' section. This section contains three radio buttons: 'Auto' (which is selected), 'Every Line', and 'Regex'. At the bottom right of the dialog are two buttons: 'Cancel' and 'Save'.

*Figure 11-1. Custom Sourcetype:test*

7. In Events Breaks, there are three types of event-breaking policies: Auto, Every Line, and Regex. Select according to your data.
8. In the Timestamp tab, there are four policies to determine timestamps for incoming data: Auto, Current Time, Advanced, and Configuration. Select according to your data.

Under Advanced, add `TIME_FORMAT`, `MAX_TIMESTAMP`, `TRUNCATE`. To change the value of `LINE_BREAKER`, refer to the following code block for `test.txt`.

```
SHOULD_LINEMERGE=false
LINE_BREAKER=([\r\n]+)\d+\s+\\"$EIT\,
NO_BINARY_CHECK=true
category=Custom
TIME_FORMAT = %m/%d/%Y %k:%M
MAX_TIMESTAMP_LOOKAHEAD = 15
TRUNCATE = 99999
```

## Custom Source Types Using `props.conf`

Creating a customized source type is the easiest and best option when using a `props.conf` file. Go to `$SPLUNK_HOME/etc/apps/$APP_NAME/local/props.conf`.

If `props.conf` does not exist, create it. You can modify the `props.conf` stanza with `[my_custom_sourcetype_name]`.

There are six crucial policies to define any custom source type in Splunk. Many policies are added and subtracted, totally depending on the input data, but overall, these six policies are important.

The policies are defined in the following code block.

```
SHOULD_LINE_MERGE=false|true
TIME_PREFIX=<regular expression>
TIME_FORMAT=<strptime-style format>
MAX_TIMESTAMP_LOOKAHEAD=<Integer>
LINE_BREAKER=<regular expression>
TRUNCATE=<Integer>
```

To create a customized source type for test1.txt, where the source type is named [Test9], follow these steps in props.conf.

```
[Test9]
TIME_PREFIX=\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\s\-\s\d{5}\s+
TIME_FORMAT = %m/%d/%Y %k:%M
MAX_TIMESTAMP_LOOKAHEAD = 15
LINE_BREAKER = ([\r\n]+)\d+\s+\\"$EIT\,
SHOULD_LINEMERGE = false
TRUNCATE = 99999
```

If you set the source type in inputs.conf for a given source, you cannot override the source type value in props.conf.

You have completed the creation of source types using fine-tuning input and the use of props.conf.

---

**Note** If you create source types in Splunk Cloud using Splunk Web, Splunk Cloud manages the source type configurations automatically. However, if you have Splunk Enterprise and manage a distributed configuration, you must distribute a new source type.

---

The next section discusses the process of anonymizing sensitive data and credentials.

## Anonymizing the Data

It is essential to mask sensitive personal information such as credit card numbers and social security numbers. You can anonymize parts of data in events to protect privacy while providing the remaining data for analysis.

There are two ways to anonymize data in Splunk.

- Use props.conf to anonymize data with a sed script
- Use props.conf and transforms.conf to anonymize data with regular expressions

## props.conf to Anonymize Data with a sed Script

In this process, you directly use props.conf to anonymize parts of data in events to protect privacy while providing the remaining data for analysis. The data can be anonymized using a sed (stream editor) script to replace a substitute string in an event. Sed is a \*nix utility that reads a file and modifies the input based on commands within or arguments supplied to the utility. You can use sed like syntax in props.conf to anonymize data.

### Syntax to Anonymize Data with a sed Script

```
SEDCMD-<class> = y/<string1>/<string2>/
```

In Test.txt, there is data that looks similar to the data in the following code block.

```
1 "$EIT,907409,38550,E,0,,0,1,0,0,16777317,0,8,A,19.079747,72.849640,65529,
195,183023,261218,23,1,15,0,4208,1148,0,1,0,0,0,4.2,E10.21,0" 27.97.83.90 -
58840 12/27/2018 0:00
```

You need to mask the IP address to XXXX. You need to first modify inputs.conf, which has a few prerequisite policies similar to the policy in the following code block.

#### inputs.conf

```
[monitor://desktop/Test1.txt]
sourcetype=Test9
```

After updating inputs.conf, you can update props.conf. In props.conf, write SEDCMD to mask data. The SEDCMD command to mask IPs is similar to the following policy.

```
SEDCMD-fixsite=s/(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/ip=xxxx/g
```

After the SEDCMD command, the props.conf for the Test9 source type is similar to the following block.

```
[Test9]
TIME_PREFIX=\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\s\-\s\d{5}\s+
TIME_FORMAT = %m/%d/%Y %k:%M
MAX_TIMESTAMP_LOOKAHEAD = 15
LINE_BREAKER = ([\r\n]+)\d+\s+"\$EIT\,
```

```
SHOULD_LINEMERGE = false
TRUNCATE = 99999
SEDCMD-fixsite=s/(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/ip=xxxx/g
```

Once the props.conf is updated for the Test9 source type, the IP address is masked. The following code block shows the Test9 data.

```
"$EIT,907409,38550,E,0,,0,1,0,0,16777317,0,8,A,19.079747,72.849640,65529,195,183023,261218,23,1,15,0,4208,1148,0,1,0,0,0,4.2,E10.21,0" ip=xxxx - 58840 12/27/2018 0:00
```

In props.conf file, the sedcmd command anonymizes the data.

Let's look at how to anonymize data using props.conf and transforms.conf with regular expressions.

## props.conf and transforms.conf to Anonymize Data with Regular Expressions

Let's use props.conf and transforms.conf to anonymize parts of data in events to protect privacy while providing remaining data. transforms.conf and props.conf transform events if the input event matches a regular expression.

In Test.txt, the data looks like the data in the following code block.

```
1 "$EIT,907409,38550,E,0,,0,1,0,0,16777317,0,8,A,19.079747,72.849640,65529,195,183023,261218,23,1,15,0,4208,1148,0,1,0,0,0,4.2,E10.21,0" 27.97.83.90 - 58840 12/27/2018 0:00
```

You mask the entire data from the IP address to XXXX. First, you need to modify inputs.conf. It has a few prerequisite policies similar to the policy in the following code block.

### inputs.conf

```
[monitor://desktop/Test1.txt]
sourcetype=Test9
```

After updating `inputs.conf`, you can update `props.conf` to call `transforms.conf` for masking data. The `props.conf` block looks similar to the following block.

```
[Test9]
TIME_PREFIX=\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\s\-\s\d{5}\s+
TIME_FORMAT = %m/%d/%Y %k:%M
MAX_TIMESTAMP_LOOKAHEAD = 15
LINE_BREAKER = ([\r\n]+)\d+\s+\\"$EIT\,
SHOULD_LINEMERGE = false
TRUNCATE = 99999
TRANSFORMS-anonymize = ip-anonymizer
```

After updating `props.conf`, you need to update `transforms.conf` so that its block is similar to the following block.

```
[ip-anonymizer]
REGEX = (?ms)(.*)(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})(.*)(?ms)
FORMAT = $1*****
DEST_KEY = _raw
```

After updating `transforms.conf`, the IP address is masked for the data that comes in the `Test9` source type.

The following code block provides the respective data.

```
1 "$EIT,907409,38550,E,0,,0,1,0,0,16777317,0,8,A,19.079747,72.849640,65529,
195,183023,261218,23,1,15,0,4208,1148,0,1,0,0,0,4.2,E10.21,0" *****
```

In `transforms.conf`, the `regex` command anonymizes the data.

Now, let's turn to merging logic in Splunk.

## Understanding Merging Logic in Splunk

In Splunk, there are various configuration files located in different places. With merging logic, all these configuration files are brought together and combined in one global file. In this section, you learn how merging is performed and get to know how Splunk combines configuration files.

## Configuration File Precedence

Splunk software uses configuration files to determine every aspect of its behavior. Configuration file precedence merges the settings from all copies of the file. Different copies have a conflicting attribute value. The file with the highest priority determines the configuration file's priority by location in the directory structure.

### Splunk Determine Precedence Order

Splunk determines the precedence order based on the following rules.

- System local directory—highest priority
- App local directories
- App default directories
- System default directory—lowest priority

### Splunk .conf Files Location

Splunk configuration files reside in `$SPLUNK_HOME/etc`.

The following directories contain Splunk configuration files.

- **`$SPLUNK_HOME/etc/system/default`**: This directory contains the default configuration files in Splunk. The default directory contains pre-configured versions of the configuration files. Never edit default files because they will be overwritten during an upgrade.
- **`$SPLUNK_HOME/etc/system/local`**: This directory contains the local configuration files. You can create and edit your files in one of these directories. These directories are not overwritten during updates.
- **`$SPLUNK_HOME/etc/apps/$app_name/local`**: This directory contains the local configuration for applications in Splunk. All configurations for your application should be placed here.



- **\$SPLUNK\_HOME/etc/users:** This is the user-specific configuration directory. If you want to change the password or any user-specific configuration, you update it here.
- **\$SPLUNK\_HOME/etc/system/README:** This directory contains the reference files. Reference files are either .example files or .spec files.

## Configuration Merging Logic

Configuration merging logic is simple. To merge, the configurations' names, stanza names, and attribute names should match.

Merging in Splunk is done based on configuration file precedence.

### Example 1: Configuration Merging (No Conflict)

Let's look at an example of configuration merging logic with no conflicts.

You have \$SPLUNK\_HOME/etc/system/local/sample1.conf.

```
[N]  
k=1  
d=2
```

And you have \$SPLUNK\_HOME/etc/system/local/sample2.conf.

```
[N]  
e=5
```

The resulting configuration has no conflicts, so Splunk merges the files.

```
[N]  
k=1  
d=2  
e=5
```

### Example 2: Configuration Merging (Conflict)

The following is for \$SPLUNK\_HOME/etc/system/local/sample1.conf.

```
[N]  
k=1  
d=2
```

You now merge with `$(SPLUNK_HOME)/etc/apps/search/local/sample1.conf`, which has the following configuration.

```
[N]
k=5
```

The resulting configuration, `$(SPLUNK_HOME)/etc/apps/search/local/sample1.conf`, is unable to overwrite `$(SPLUNK_HOME)/etc/system/local/sample1.conf` because in file precedence, files in the system (`$(SPLUNK_HOME)/etc/system/local/sample1.conf`) have greater priority than files in apps (`$(SPLUNK_HOME)/etc/apps/search/local/sample1.conf`). The resulting configuration is as follows.

```
[N]
k=1
d=2
```

### Example 3: Configuration Merging (Conflict)

You have `$(SPLUNK_HOME)/etc/apps/search/default/sample1.conf`.

```
[N]
k=1
d=2
```

And you want to merge `$(SPLUNK_HOME)/etc/apps/search/local/sample1.conf`.

```
[N]
k=5
```

The resulting configuration, `$(SPLUNK_HOME)/etc/apps/search/local/sample1.conf`, can overwrite the value of `$(SPLUNK_HOME)/etc/apps/search/default/sample1.conf` because in file precedence, the files in local have greater priority than files in default. Hence, the resulting configuration is as follows.

```
[N]
k=5
d=2
```

Let's now discuss debugging a configuration file using Btool.

## Debugging Configuration Files

Btool in Splunk is used for displaying merged on-disk configurations. Btool is a command-line tool that can help troubleshoot configuration file issues or see what values are being used by your Splunk Enterprise installation. Splunk provides Btool in the `$SPLUNK_HOME/bin` file. Btool displays the on-disk configuration file settings, and if you change a setting, then you do not need to restart Splunk.

The syntax for `btool <conf_file_prefix>` is as follows.

```
./splunk cmd btool <conf_file_prefix> list
```

In the preceding syntax, `<conf_file_prefix>` stands for the name of the configuration you are interested in.

The syntax for `btool <conf_file_prefix>` is as follows.

```
./splunk cmd btool server list --debug | grep '\['
```

In the preceding syntax, `./splunk cmd btool server list --debug | grep '['` determines which `server.conf` stanzas are being recognized.

### Example: Btool for Troubleshooting a Configuration File

This is the configuration of `$SPLUNK_HOME/etc/apps/test/local/inputs.conf`.

```
[monitor://desktop/Test1.txt]
sourcetype=Test9
```

Change the configuration to the following code.

```
[monitor://desktop/Test1.txt]
ourcetype=Test9
```

To troubleshoot the configuration file, go to `$SPLUNK_HOME/bin`.

```
./splunk cmd btool check
```

Note that Btool prompts you with an error `invalid key in the stanza`. A similar error is shown in Figure 11-2.

```
(base) Deeps-MacBook-Air:bin deepmehta$ ./splunk cmd btool check
Invalid key in stanza [monitor://desktop/Test1.txt] in /Applications/Splunk/etc/apps/test/local/inputs.conf, line 2: ourcetype (value: Test9).
```

**Figure 11-2.** *Btool error check for sourcetype:Test9*

Let's now look at creating a diag.

## Creating a Diag

In Splunk, a diag is used for collecting basic information regarding your Splunk environment, instance, configuration file, and so forth. It gathers information regarding server specification, OS version, file system, and current open connections from the node running in the Splunk environment. A diag can contain app configurations, internal Splunk log files, and index metadata from the Splunk instance. Diags are stored in `$SPLUNK_HOME/var/run/diags`.

## Creating a Diag in Splunk

Diags can be created using Splunk Web or a terminal/CLI. Starting in Splunk 6.0, you can generate diag for remote instances, but you need to have at least one of the following server roles.

- A search head, which is the only search head in a deployment
- A clustered search head
- A clustered indexer
- An indexer cluster master

To create a diag using the command line, go to `$SPLUNK_HOME/bin`.

- In Linux, run the following code.

```
./splunk diag
```

- In Windows, run the following code.

```
splunk diag
```

Figure 11-3 shows the diag created for a local instance.

```
(base) Deeps-MacBook-Air:~$ sudo ./splunk diag
Collecting components: conf_replication_summary, consensus, dispatch, etc, file_validate, index_files, index_listing, kvstore, log, searchpeers, suppression_listing
Skipping components: rest
Selected diag name of: diag-Deeps-MacBook-Air.local-2019-11-16_18-39-31
Starting splunk diag...
Logged search filtering is enabled.
Skipping REST endpoint gathering...
Determining diag-launching user...
Getting version info...
Getting system version info...
Getting file integrity info...
Getting network interface config info...
Getting splunk processes info...
Getting netstat output...
Getting info about memory, ulimits, cpu (on windows this takes a while)...
Getting etc/auth filenames...
Getting Sirenhole filenames...
Getting search peer bundles listings...
Getting conf replication summary listings...
Getting suppression files listings...
Getting KV Store listings...
Getting Index listings...
Copying Splunk configuration files...
filtered out file '/Applications/Splunk/etc/apps/splunk_archiver/java-bin/jars/vendors/spark/2.3.3/lib/spark-core_2.11-2.3.3.jar' limit: 10485760 size: 13123587
filtered out file '/Applications/Splunk/etc/apps/splunk_archiver/java-bin/jars/thirdparty/hive_1.2/hive-exec-1.2.1.jar' limit: 10485760 size: 20599029
filtered out file '/Applications/Splunk/etc/apps/splunk_archiver/java-bin/jars/thirdparty/aws/aws-java-sdk-1.10.8.jar' limit: 10485760 size: 21806573
The following certificates were excluded from the diag output automatically.
/Applications/Splunk/etc/auth/appsCA.pem
/Applications/Splunk/etc/auth/server.pem
/Applications/Splunk/etc/auth/cloudCA.pem
/Applications/Splunk/etc/auth/applicenseCA.pem
/Applications/Splunk/etc/auth/ca.pem
/Applications/Splunk/etc/auth/cacert.pem
/Applications/Splunk/etc/auth/distServerKeys/trusted.pem
/Applications/Splunk/etc/auth/distServerKeys/private.pem
/Applications/Splunk/etc/auth/audit/public.pem
/Applications/Splunk/etc/auth/audit/private.pem
/Applications/Splunk/etc/auth/splunkweb/cert.pem
/Applications/Splunk/etc/auth/splunkweb/privkey.pem
If you have any certs that were not auto-detected, please add them to an EXCLUDE rule in the [diag] stanza of server.conf.
Copying Splunk log files...
Copying bucket info files...
Copying Splunk dispatch files...
Copying Splunk consensus files...
Adding manifest files...
Adding cachemanager_upload.json...
Cleaning up...
Splunk diagnosis file created: /Applications/Splunk/diag-Deeps-MacBook-Air.local-2019-11-16_18-39-31.tar.gz
```

**Figure 11-3.** Splunk Diag

To exclude the file from diag, run the following code.

- In Linux
 

```
./splunk diag --exclude "*/passwd"
```
- In Windows
 

```
splunk diag --exclude "*/passwd"
```

When a Splunk diag is run, it produces a tar.gz file and a diag.log file.

You have now reached the end of the chapter. You can check your knowledge using the questions provided after the Summary.

## Summary

This chapter discussed the various configuration files existing in Splunk: `inputs.conf`, `outputs.conf`, `props.conf`, `transforms.conf`, `deploymentclient.conf`, and `indexes.conf`. You learned about their key/value attributes and their utility in creating and modifying various commands. These files are responsible for storing enterprise and app configuration information. You also learned about custom source type creation using both Splunk Web and `props.conf`. You learned the process of anonymizing the sensitive or private data and the process of merging the configurational files located in several places into a single global file. In the end, you learned about debugging configuration files and the process of creating a `diag` that collects basic data like the environment instance, server specification, and open connections from the nodes. The chapter focused on configuration files, which are a crucial element of Splunk Enterprise.

According to Splunk Admin Exam Blueprint configuration files i.e. Module 16 accounts to 10%, custom source type i.e. Module 15 accounts to 5%, anonymizing of the data i.e. Module 17 accounts to 5% and merging of the configuration file i.e. Module 1 to 5%.

The next chapter is a Splunk admin mock exam to help you get an a better idea of the Splunk admin exam patterns.

## Multiple-Choice Questions

- A. Default configuration files in Splunk are saved in which location?
  1. `$SPLUNK_HOME/etc/system/local`
  2. `$SPLUNK_HOME/etc/bin/default`
  3. `$SPLUNK_HOME/etc/bin/local`
  4. `$SPLUNK_HOME/etc/system/default`
- B. The `props.conf` file applies rules to regular expressions.
  1. true
  2. false

- C. Where does the inputs.conf file reside? (Select all that apply.)
  - 1. universal forwarder
  - 2. heavy forwarder
  - 3. search head
  - 4. indexer
  - 5. light forwarder
  - 6. none of the above
  
- D. If you set the source type in inputs.conf for a given source, you cannot override the source type value in props.conf.
  - 1. true
  - 2. false
  
- E. In which Splunk configuration is the SEDCMD command used?
  - 1. props.conf
  - 2. inputs.conf
  - 3. transforms.conf
  - 4. outputs.conf
  
- F. According to Splunk merging logic, which file has the highest priority?
  - 1. system default directory
  - 2. app local directory
  - 3. app default directory
  - 4. system local directory

## Answers

- A. 4
- B. 2
- C. 1, 2, 3, 4
- D. 1
- E. 1
- F. 4

## Reference

For more information, go to <https://docs.splunk.com/Documentation/Splunk/latest/Data/Anonymizedata>.



## CHAPTER 12

# Splunk Admin Exam Set

This chapter presents multiple-choice questions useful for Splunk Enterprise Administrator certification to give you an idea of what appears on the exams.

## Questions

- A. Default configuration files in Splunk are saved in which location?
  1. \$SPLUNK\_HOME/etc/system/local
  2. \$SPLUNK\_HOME/etc/bin/default
  3. \$SPLUNK\_HOME/etc/bin/local
  4. \$SPLUNK\_HOME/etc/system/default
- B. Which of the following statements describe deployment management? (Select all that apply.)
  1. requires an Enterprise license
  2. is responsible for sending apps to forwarders
  3. once used, is the only way to manage forwarders
  4. can automatically restart the host OS running the forwarder
- C. For a single line event source, it is most efficient to set SHOULD\_linemerge to what value?
  1. true
  2. false
  3. <regex string>
  4. newline character

- D. The universal forwarder has which capabilities when sending data? (Select all that apply.)
  - 1. sending alerts
  - 2. compressing data
  - 3. obfuscating/hiding data
  - 4. indexer acknowledgement
  
- E. When configuring monitor inputs with whitelists or blacklists, what is the supported method for filtering lists?
  - 1. slash notation
  - 2. regular expressions
  - 3. irregular expressions
  - 4. wildcards only
  
- F. Which hardware attribute needs to be changed to increase the number of simultaneous searches (ad hoc and scheduled) on a single search head?
  - 1. disk
  - 2. CPUs
  - 3. memory
  - 4. network interface cards
  
- G. Which valid bucket types are searchable? (Select all that apply.)
  - 1. hot buckets
  - 2. cold buckets
  - 3. warm buckets
  - 4. frozen buckets

- H. Which of the following are supported options when configuring optional network inputs?
1. metadata override, sender filtering options, network input queues (quantum queues)
  2. metadata override, sender filtering options, network input queues (memory/persistent queues)
  3. filename override, sender filtering options, network output queues (memory/persistent queues)
  4. metadata override, receiver filtering options, network input queues (memory/persistent queues)
- I. Which Splunk forwarder type allows parsing of data before forwarding to an indexer?
1. universal forwarder
  2. parsing forwarder
  3. heavy forwarder
  4. advanced forwarder
- J. What is the default character encoding used by Splunk during the input phase?
1. UTF-8
  2. UTF-16
  3. EBCDIC
  4. ISO 8859
- K. How does the Monitoring Console monitor forwarders?
1. by pulling internal logs from forwarders
  2. by using the forwarder monitoring add-on
  3. with internal logs forwarded by forwarders
  4. with internal logs forwarded by the deployment server

- L. In which scenario would a Splunk administrator want to enable data integrity check when creating an index?
  - 1. to ensure that hot buckets are still open for writes and have not been forced to roll to a cold state
  - 2. to ensure that configuration files have not been tampered with for auditing and/or legal purposes
  - 3. to ensure that user passwords have not been tampered with for auditing and/or legal purposes
  - 4. to ensure that data has not been tampered with for auditing and/or legal purposes
  
- M. Which Splunk component distributes apps and certain other configuration updates to search head cluster members?
  - 1. deployer
  - 2. cluster master
  - 3. deployment server
  - 4. search head cluster master
  
- N. You update a props.conf file when Splunk is running. You do not restart Splunk and you run the following command.
  - 1. a list of all the configurations on disk that Splunk contains
  - 2. a verbose list of all configurations as they were when splunkd started
  - 3. a list of props. conf configurations as they are on-disk, along with a file path from which the configuration is located
  - 4. a list of the current running props, conf configurations and a file path from which the configuration was made

- O. Which of the following are methods for adding inputs in Splunk? (Select all that apply.)
1. CLI
  2. Splunk Web
  3. editing inputs.conf
  4. editing monitor.conf
- P. Local user accounts created in Splunk store passwords in which file?
1. \$ SFLUNK\_KOME/etc/passwd
  2. \$ SFLUNK\_KCME/etc/authentication
  3. \$ S?LUNK\_HCME/etc/users/passwd.conf
  4. \$ SPLUNK HCME/etc/users/authentication.conf
- Q. What are the minimum required settings when creating a network input in Splunk?
1. protocol, port number
  2. protocol, port, location
  3. protocol, username, port
  4. protocol, IP, port number
- R. Which Splunk indexer operating system platform is supported when sending logs from a Windows universal forwarder?
1. any OS platform
  2. Linux platform only
  3. Windows platform only
  4. none of the above
- S. Which of the following authentication types requires scripting in Splunk?
1. ADFS
  2. LDAP
  3. SAML
  4. RADIUS

**Answers**

- A. 4
- B. 2
- C. 2
- D. 2, 4
- E. 2
- F. 2
- G. 1, 2, 3
- H. 2
- I. 3
- J. 1
- K. 3
- L. 4
- M. 1
- N. 3
- O. 1, 2, 3
- P. 4
- Q. 1
- R. 4
- S. 1, 2, 3

## Summary

This chapter focused on multiple-choice questions useful for admin certification. You have come to the end of Module 2, which covered all the chapters that address Splunk admin certification. The next chapter begins the discussion on advanced Splunk deployment.

**PART III**

# **Advanced Splunk**

## CHAPTER 13

# Infrastructure Planning with Indexer and Search Head Clustering

This module deals with the advanced Splunk deployment. Deployment acts as a centralized configuration manager for instances; network-based installations do not require administrators to individually install each operating system.

The first chapter in this module discusses the architecture of Splunk. It also covers clustering. The following topics are discussed in this chapter.

- Capacity planning
- Configuring a search peer
- Configuring a search head
- Search head clustering
- Multisite indexer clustering
- Designing Splunk architecture

By the end of this chapter, you will have learned about designing architecture and resource planning in Splunk and be able to implement index clustering and search head clustering. By the time you complete this chapter, you will have covered 32% of the Splunk architect exam blueprint.



## Capacity Planning for Splunk Enterprise

Capacity planning plays a crucial role in scaling Splunk Enterprise. The components for capacity planning in Splunk Enterprise are as follows.

- An **indexer** is an instance that indexes local data. The indexer provides data processing and storage for remote and local data.
- A **search head** handles search management functions. It redirects search requests to peers and merges the result once the peer node processes the request.
- A **forwarder** is an instance that forwards data to indexers for data processing and storage.

## Dimensions of a Splunk Enterprise Deployment

Capacity planning in Splunk Enterprise deployment has various dimensions that affect it.

- **Incoming data:** The greater the amount of data you send to Splunk, the greater the time and number of resources needed to process the incoming data.
- **Indexed data:** As the data in Splunk Enterprise increases, you need more I/O bandwidth to store data.
- **Concurrent users:** As more users use a Splunk instance at the same time, the instances require more resources to perform searches and create reports and dashboards.
- **Saved searches:** Splunk needs the capacity to run saved searches efficiently.

You start with the incoming data's effect on capacity planning.

## Incoming Data Affects Splunk Enterprise Performance

Incoming data plays a critical role in capacity planning in Splunk. If there is a lot of data going to the indexer, but the physical memory is not large enough to process this data, indexer performance slows down. As data in Splunk increases, more system memory is utilized. Therefore, the amount of incoming data needs to be considered during capacity planning.

## Indexed Data Affects Splunk Enterprise Performance

Indexed data also plays a critical role in capacity planning. Splunk Enterprise captures incoming data and gives it to the indexer. As the index increases, the available space decreases, and as the indexed data increases, searches slow down processing.

## Concurrent Users Affects Splunk Enterprise Performance

Concurrent users greatly affect Splunk capacity planning. An indexer needs to provide a CPU cores for every search that the users invoke. When multiple users are running searches, all the available CPU cores are quickly exhausted.

## Saved Searches on Splunk Enterprise Performance

Concurrent users affect Splunk capacity planning. Saved searches, on average, consume about 1 CPU core and a fixed amount of memory. A saved search increases I/O when the disk system looks in the indexer to fetch data. If you schedule too many simultaneous saved searches, it leads to an exhaustion of resources.

## Disk Storage for Splunk Enterprise

Splunk calculates disk storage based on the following formula.

( Daily average indexing rate ) × ( retention policy ) × 1/2

- **Daily average indexing rate** stats incoming data in Splunk Enterprise.
- A **retention policy** is the parameter set in `indexes.conf` called `FrozenTimePeriodInSecs`.

Ideally, Splunk Enterprise stores raw data at approximately half its original size using compression and a lexicographical table. If Splunk Enterprise contains 150 GB of usable disk space, you can store nearly a hundred days' worth of data at an indexing rate of 3 GB/day.

Let's now move forward to learn about the three processes of configuring the search head and search peer.

## Configuring a Search Peer

A Splunk Enterprise instance can run both as a search head and a search peer. A search peer can perform indexing and respond to search incoming requests from search heads.

Let's first discuss the process of configuring a search peer using Splunk Web.

### Configuring a Search Peer from Splunk Web

To configure a search peer using Splunk Web, refer to the following steps.

1. Go to settings using Splunk Web and go to Distributed Search.
2. Move to Search peers and click Add New.
3. Enter the IP address or the hostname of the indexer and enter **mgmt port-no** followed by a colon (*hostname:mgmt\_port*, *IPaddress:mgmt\_port*) for authentication to share a public key between the search head and indexer. Please provide the username and password of the remote indexer on the search head. (Refer to Figure 13-1.)

**Add search peers**

Use this page to explicitly add distributed search peers. Enable distributed search through the Distributed search setup page in Splunk Settings.

Peer URI \*

Specify the search peer as `servername:mgmt_port` or `URI:mgmt_port`. You must prefix the URI with its scheme. For example: `https://spl.example.com:8089`.

**Distributed search authentication**

To share a public key for distributed authentication, enter a username and password for an admin user on the remote search peer.

Remote username \*

Remote password \*

Confirm password

**Figure 13-1.** Configure Search Peer

## Configure Splunk Search Peer from the .conf File

To configure a Splunk search peer by editing the .conf file, refer to the following steps.

1. In the search head, create or edit `distsearch.conf` located in `$Splunk_Home$/etc/local` or `%Splunk_Home$/etc/local%`.
2. Add the following stanza in `distsearch.conf`.

```
[distributedSearch]
servers = <IPAddress1>:<mgmt_port>,<IPAddress2>:<mgmt_port>,
<IPAddress3>:<mgmt_port3>,. . . . .<IPaddressn>:<mgmt_port>
```

- **servers** is the address of the distributed search group server.

If you add a search head by editing .conf files, you need to distribute keys manually, as follows.

1. Copy the keys of search head file from `$SPLUNK_HOME/etc/auth/distServerKeys/trusted.pem` to indexer location `$SPLUNK_HOME/etc/auth/distServerKeys/<searchhead_name>/trusted.pem`.
2. The `<searchhead_name>` is the search head name, which is configured in `server.conf` of the search head.
3. Restart the Splunk instance.

## Configure Search Peer from Splunk CLI

To configure a search peer using Splunk CLI, refer to the following steps.

```
splunk add search-server (IPAddress:mgmt_port or hostname:mgmt_port)
-auth<user>:<password> -remoteUsername(username) -remotePassword(password)
```

- **search-server** is the address of the remote server.
- **-auth** is the username and password of the current instance.

- **-remoteUsername** is the username of the remote instance used for authentication.
- **-remotePassword** is the password of a remote instance used for authentication.

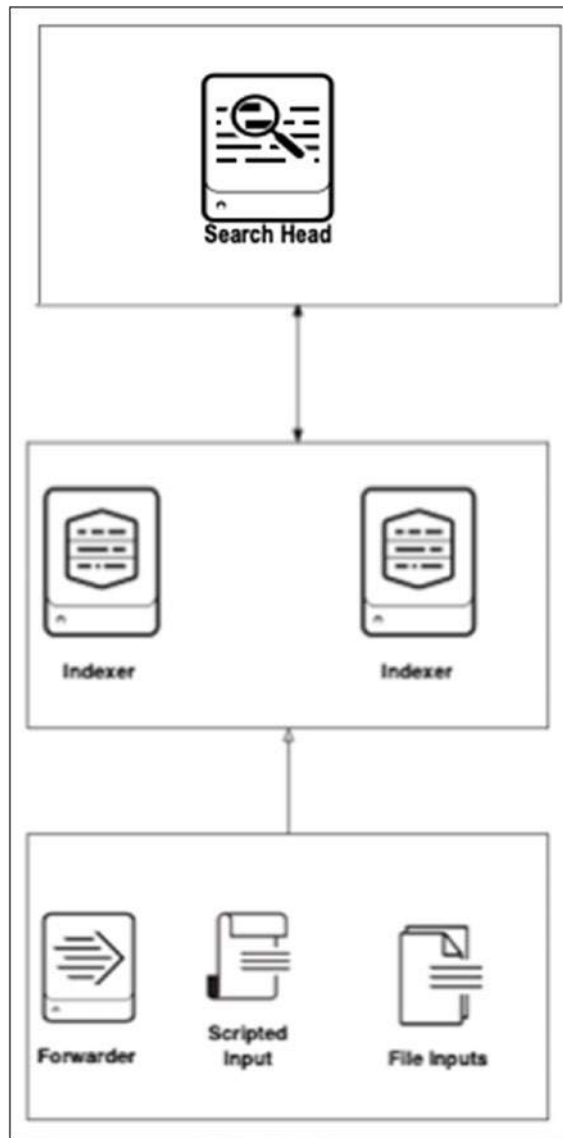
You have now covered the entire process of configuration methods used for the search peer.

The next section discusses search head clustering.

## Configure a Search Head

A search head can perform searching but not indexing, a process known as a dedicated search head. Normally in a distributed environment, the instances that manage search management functions generally direct search requests to the search peers. The search head directly communicates with the indexer.

Figure 13-2 shows the constituent structure of a Splunk search head.



**Figure 13-2.** *Search Head Constituent Architecture*

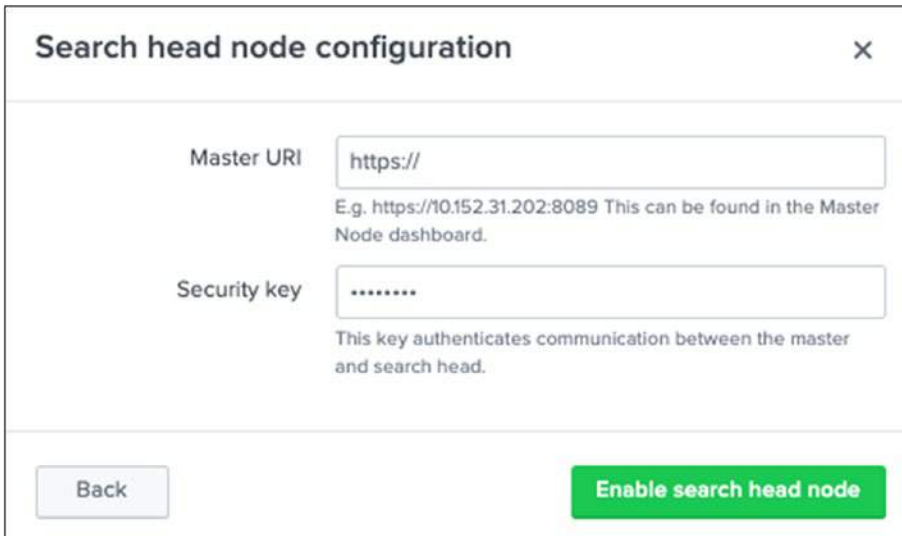
The CPU needs to be changed to increase the number of simultaneous searches (ad hoc and scheduled) on a single search head because ad hoc searches consume one core and increase I/O per execution. If you don't increase CPU, the Splunk system won't run simultaneous searches.

Let's now discuss the process of configuring a search head using Splunk Web.

## Configuring a Search Head Using Splunk Web

To configure a search head using Splunk Web, refer to the following steps.

1. Go to Settings and then go to the Distributed Search panel.
2. Move to Index Clustering ► Search Head Node, and click Add New.
3. Enter the indexer’s IP address or the hostname. Enter **mgmt port-no** followed by a colon (*hostname:mgmt\_port*, *IPaddress:mgmt\_port*) for authentication to share a public key between the search head and indexer. Please provide the common secret key of the master node (see Figure 13-3).



**Figure 13-3.** *Configure Search Head*

## Configure Splunk Search Head Using .conf file

To configure the search head clustering by editing the .conf file directly, refer to the following steps.

1. In the search head in Splunk, create or edit the `server.conf` located in `$Splunk_Home$/etc/local` or `%Splunk_Home$/etc/local%`.

2. Edit the following stanza in the server.conf files. Refer to the following stanza.

```
[clustering]
master_uri =(IPAddress:mgmt_port or hostname:mgmt_port)
mode = searchhead
pass4SymmKey = <key>
```

- **master\_uri** is the address of the master server.
- **mode** is the type of clustering mode.
- **pass4SymmKey** is the cluster key.

After updating the configuration file, restart the Splunk instance.

## Configuring a Search Head from Splunk CLI

To configure a search head using the Splunk CLI, use the following.

```
splunk edit cluster-config -mode searchhead -master_uri(IPAddress:mgmt_port
or hostname:mgmt_port) -secret your_key
```

```
splunk restart
```

- **-master\_uri** is the address of the master server.
- **-secret** is the cluster secret key.

## Search Head Clustering

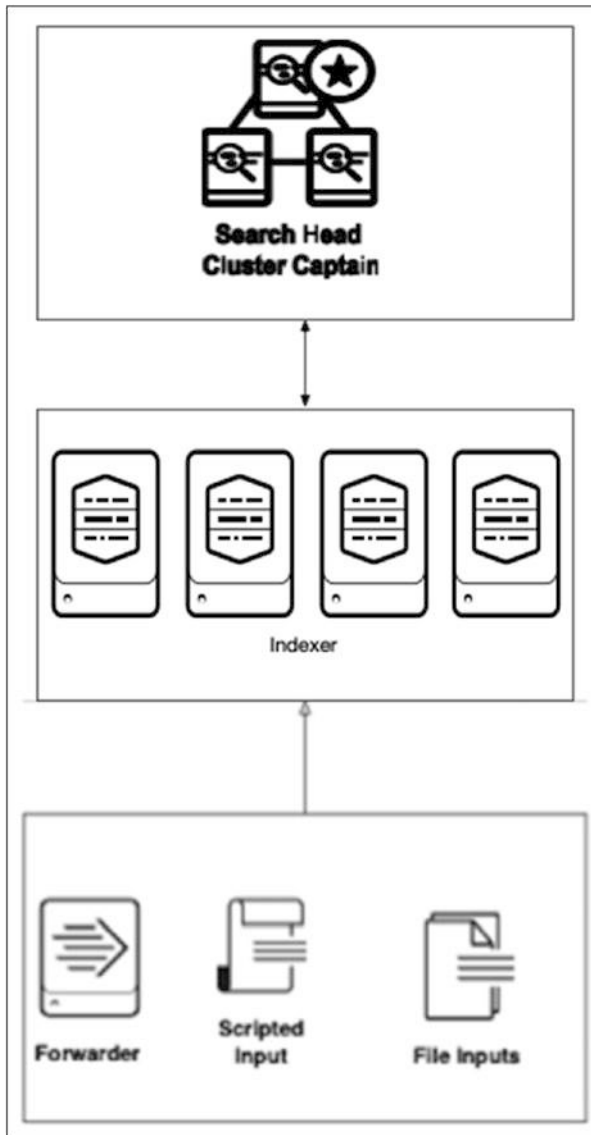
A search head cluster consists of a group of search heads that share configurations, scheduling jobs, and search artifacts. It consists of a search head cluster captain so that the entire cluster is more consistent. The search head captain can be any of the search head cluster members and is selected by the captain election.

- A **dynamic captain** is the elected search head captain from the search head cluster members. The dynamic captain can be changed from time to time and is generally used in a cluster that is working normally.



- A **static captain** is not elected through captain elections but is made to overcome disaster recovery, network issues, and so forth. A static captain is made only when the election captain is unable to make a new captain. It is temporary.

The search head clustering architecture is shown in Figure 13-4.



**Figure 13-4.** Search Head Clustering Constituent Architecture

## Search Head Cluster Captain

The captain is the search head cluster member that performs extra functions apart from search activities. It coordinates the activity of the cluster. You are able to make your search head run either scheduled searches or ad hoc searches. If your current captain fails, then a new captain is selected after the election. The new captain can be any Splunk search head cluster member.

To configure cluster members to run ad hoc searches, edit `server.conf`.

```
[shclustering]
adhoc_searchhead = true
```

To configure captain to run ad hoc searches, edit `server.conf`.

```
[shclustering]
captain_is_adhoc_searchhead = true
```

Let's now look at the captains' roles and functions.

## The Role of Captains

The following are a captain's role in a Splunk search head cluster.

- Scheduling jobs
- Coordinating alerts and alert suppressions across the cluster
- Pushing the knowledge bundle to search peers
- Coordinating artifact replication
- Replicating configuration updates

## Captain Election

A search head cluster uses a dynamic captain. When a condition arises, the cluster conducts an election, and any Splunk search head member can become a Splunk head captain.

Captain election happens only when

- the current captain fails
- there are network issues

- the current captain steps down because it is not able to detect that major members are participating in the cluster
- network partition occurs

## Configure Search Head Cluster Using CLI in Splunk

To configure the search head clustering from Splunk CLI, do the following.

```
splunk init shcluster-config -auth<username>:<password> -mgmt_
uri<ip>:<mgmt_port>/<hostname>:<mgmt_port> -replication_factor<positive_
integer> -replication_port<port> -shcluster_label<string> -secret<key>

splunk restart
```

- **-auth** parameter specifies current login credentials.
- **-mgmt\_uri** specifies the URI of the current instance.
- **-replication\_factor** specifies the number of copies of each artifact that the cluster maintains.
- **-replication\_port** specifies the port that the instance uses to listen to other artifacts. It can be any unused port.
- **-shcluster\_label** is an option used to label a cluster.
- **-secret** is the key that specifies the security key.

## Configure Dynamic Search Captain Using Splunk CLI

To configure a dynamic search captain in Splunk CLI, do the following.

```
splunk bootstrap shcluster_captain -servers_list<ip1>:<mgmt_port>/
<hostname1>:<mgmt_port>,<ip2>:<mgmt_port>/<hostname2>:<mgmt_port>,
.....,<ipn>:<mgmt_port>/<hostnamen>:<mgmt_port>

splunk restart
```

**To add minority group members to dynamic captain using Splunk CLI**

To add minority search members to dynamic search captain using the Splunk CLI command. Use captain server terminal to add shcluster members.

```
splunk add shcluster-member -new_member_uri <ip>:<mgmt_port>
```

## Configure Static Search Captain Using Splunk CLI

To configure static search captain using Splunk CLI, refer to the following steps.

```
splunk edit shcluster-config -election false -mode captain captain_uri  
<ip1>:<mgmt_port> or <hostname1> :<mgmt_port>
```

```
splunk restart
```

To add minority search members to a dynamic search captain using the Splunk CLI command, use a peer node terminal to add shcluster members.

```
splunk edit shcluster-config -election false -mode member captain_uri  
<ip1>:<mgmt_port> or <hostname1> :<mgmt_port>
```

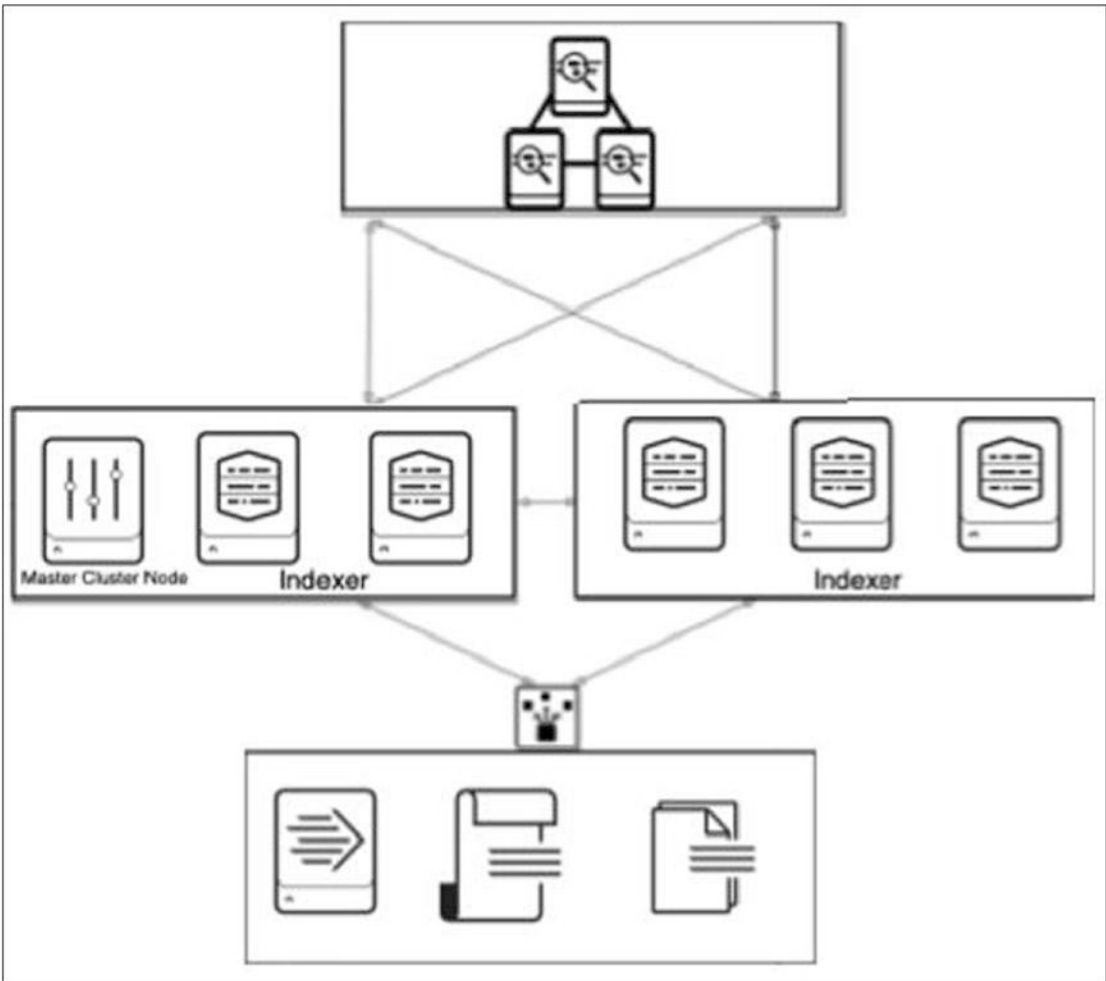
```
splunk restart
```

## Multisite Indexer Clustering

Multisite indexer clustering is an indexer cluster that consists of multiple sites. Each site needs to follow specific replication and search factor rules. Multisite clusters offer two key benefits over single site clusters.

**Improved disaster recovery:** By maintaining multiple copies of data at multiple locations, saves your data in a disaster. Multisite indexer clustering keeps multiple copies of data at multiple locations and provides site failover capabilities.

**Search affinity:** This configures each site to have local data and a full set of searchable data. The search head on each site limits searches to peer nodes only. This eliminates any need, under normal conditions, for search heads to access data on other sites, greatly reducing network traffic between sites.



**Figure 13-5.** *Multisite Index Clustering Constituent Architecture*

## Configure Multisite Indexer Clustering Using .conf Files

To configure multisite indexer clustering by editing the .conf file, refer to the following steps.

1. In the search head in Splunk, create or edit the server.conf located in \$Splunk\_Home\$/etc/local or %Splunk\_Home\$/etc/local%.
2. Edit the following stanza in the server.conf files.

```
[general]
site = site1,site2,site3,.....,site n
[clustering]
```

```

mode = master
multisite = true
available_sites = site1,site2,.....,site n
site_replication_factor = origin:<number>,total:<number>
site_search_factor = origin:<number>,total:<number>
pass4SymmKey = <key>
cluster_label = <cluster name>

```

- **Available\_sites** is used in a master node to show which sites are available for the Splunk environment.
- **Site\_replication\_factor** specifies the total copies of raw data that the cluster should maintain.
- **Site\_search\_factor** is used in a master node to maintain several searchable copies in the Splunk environment for disaster recovery.
- **pass4Symmkey** authenticates communication between nodes.
- **Cluster\_label** is the name of the cluster.

To configure a multisite indexer peer node in Splunk using .conf files, refer to the following steps.

1. In the search head in Splunk, create or edit the server.conf located in \$Splunk\_Home\$/etc/local or %Splunk\_Home\$/etc/local%.
2. Edit the following stanza in the server.conf files.

```

[general]
site = site1,site2,site3,.....site n

[replication_port://<port number>]

[clustering]
master_uri = <ip>:<port>/<host name>:<port>
mode = slave
pass4SymmKey =<key>

```

- **available\_sites** is used in a master node to show which sites are available in the Splunk environment.

- **master\_uri** is the server address.
- **pass4Symmkey** authenticates communication between nodes.

To configure the multisite indexer search head in the Splunk using .conf files. Refer to the following steps.

1. In the search head in Splunk, create or edit the server.conf file located in \$Splunk\_Home\$/etc/local or %Splunk\_Home\$/etc/local%.
2. Edit the following stanza in the server.conf files.

```
[general]
site = site1,site2,site3,....,site n

[clustering]
multisite = true
master_uri = <ip>:<port>,<host name>:<port>
mode = searchhead
pass4SymmKey = <key>
```

- **available\_sites** is used in master node to show which sites are available in the Splunk environment.
- **master\_uri** is the server address.
- **pass4Symmkey** authenticates communication between nodes.

For more information on multisite search head clustering, go to <https://docs.splunk.com/Documentation/Splunk/7.3.0/Indexer/Multisiteconf>.

## Configure Splunk Multisite Indexer Clustering Using CLI

To configure a peer node as a master node for a multisite indexing cluster in Splunk using CLI commands, refer to the following stanza.

```
splunk edit cluster-config -mode master -multisite true -available_sites
site1,site2,....,site n -site_replication_factor origin:<number>,
total:<number> -site_search_factor origin:<number>,total:<number>
-secret<key>
```

```
splunk restart
```

- **-available\_sites** is used in master node to show which sites are available in the Splunk environment.
- **-site\_replication\_factor** specifies the total copies of raw data that the cluster should maintain.
- **-site\_search\_factor** is used in the master node to maintain several searchable copies in the Splunk environment in disaster recovery.
- **-secret** authenticates communication between nodes.
- **-cluster\_label** is the name of the cluster.

To configure a peer node as slave node for a multisite indexing cluster using CLI commands, refer to the following stanza.

```
splunk edit cluster-config -mode slave -site site1,site2,site3,.....,
site n -master_uri <ip>:<port>/<hostname>:<port> -replication_port
<port number> -secret <key>
```

```
splunk restart
```

- **-available\_sites** is used in a master node to show which sites are available in the Splunk environment.
- **-master\_uri** is the server address.
- **-secret** authenticates communication between nodes.

To configure a peer node as a search head for a multisite indexing cluster using CLI commands, refer to the following stanza.

```
splunk edit cluster-config -mode searchhead -site site1,site2,...,siten
-master_uri <ip>:<port>/<hostname>:<port> -secret <key>
```

```
splunk restart
```

- **available\_sites** is used in a master node to show which sites are available in the Splunk environment.
- **master\_uri** is the server address.
- **pass4Symmkey** authenticates communication between nodes.

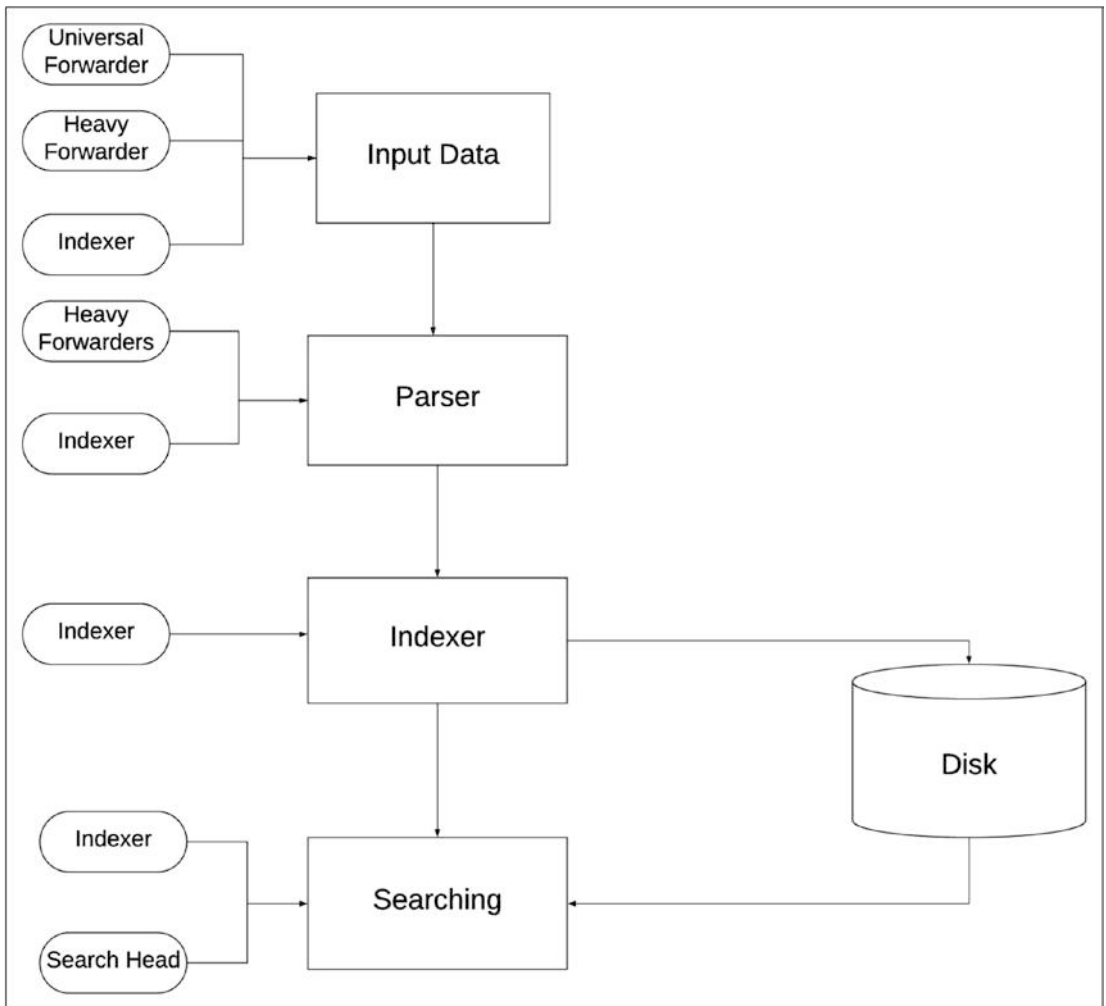


Before moving forward to discuss the design of the Splunk architecture, let's review what you have already learned in this chapter: capacity planning in the Splunk environment, configuring search heads, search head clustering, captains, and clustering the multisite indexer. The next section discusses designing and architecture in Splunk.

Designing Splunk architecture is a complex job. To design in Splunk, you should be aware of the series of operations in its architecture and all the software components.

- **Data input:** To input data in Splunk, you use a universal forwarder. However, a heavy forwarder or an indexer can get data in the Splunk environment.
- **Data parsing:** To parse data, you use a heavy forwarder and the indexer.
- **Data indexing:** To index data in a Splunk environment, you can only use the indexer.
- **Data searching:** To search data in a Splunk environment, you should use a search head, but you can also use an indexer.

The chart in Figure 13-6 depicts the entire Splunk architecture.



**Figure 13-6.** *Splunk Constituent Architecture:Components*

At this point, you have already revised the Splunk architecture with all the required components for data input, data parsing, data indexing, and searching data. Implementing Splunk architecture is not a big issue based on the diagram. But how do you design a Splunk architecture?

To understand how to design, you need to know about Splunk Validated Architectures.

## Splunk Validated Architectures (SVAs)

Splunk Validated Architectures (SVAs) were adapted to achieve a stable, efficient, and repeatable deployment. SVAs ensure platform scaling and troubleshooting on the Splunk platform and reduce the total cost of ownership.

The following are the characteristics of SVAs.

**Performance:** SVAs help organizations improve performance and helps with stability.

**Complexity:** SVAs help organizations remove complexity from your environment as complexity acts as the biggest barrier when you plan for scaling Splunk Environment.

**Efficiency:** SVAs help organizations achieve efficiency by improving operations and accelerating time to value.

**Cost:** SVAs help organizations reduce the cost of the organization by reducing the cost of ownership.

**Agility:** SVAs help organizations adapt as they scale up when data grows.

**Maintenance:** SVAs help organizations reduce the organization's maintenance efforts by channeling sources in the proper dimension.

**Scalability:** SVAs help organizations scale efficiently and seamlessly.

**Verification:** SVAs help organizations assures that the Splunk deployment is built on best practices.

## Designing Splunk Validated Architectures

When designing a Splunk architecture, you need to understand a few key points to maintain your workflow.

- **Availability:** The Splunk environment needs to be operational at any time and recover from planned and unplanned outages or disruptions.

- **Performance:** The Splunk environment needs to maintain the same level of service under different conditions.
- **Scalability:** The Splunk environment needs to be designed to scale on all tiers and handle the workload effectively.
- **Security:** The Splunk environment should protect data, configurations, and assets.
- **Manageability:** The Splunk environment is centrally operable and manages all the tiers.

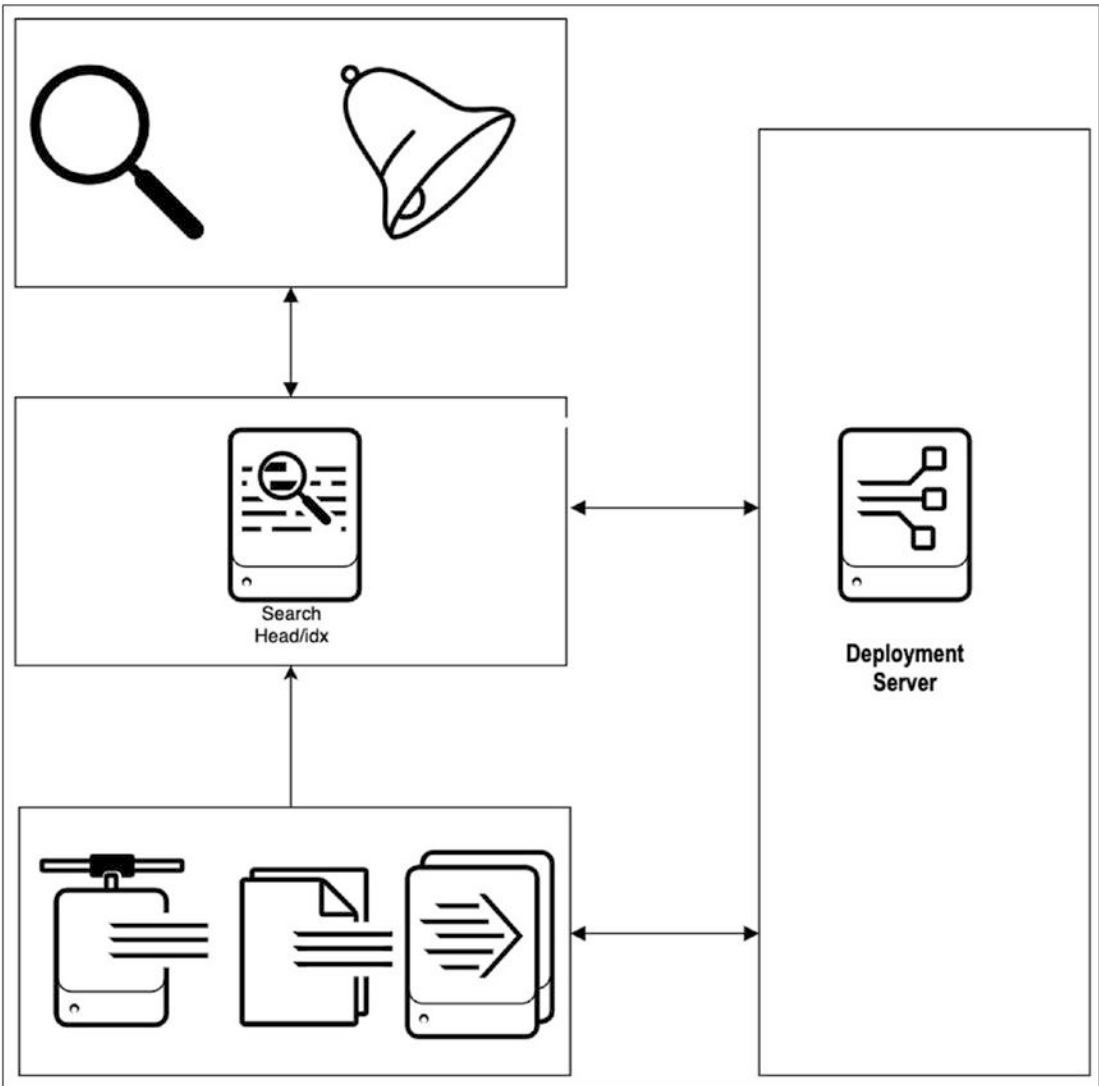
## Small-Scale Enterprise Deployment

If the indexing volume ranges between 20 and 300 GB/day, and the organization has between one and ten users, a small-scale enterprise deployment is favorable. A small-scale enterprise deployment has various forwarders to forward the data to the indexer. Indexers and search head are placed together to use the server more efficiently.

Figure 13-7 shows a small enterprise deployment.

- **Data input:** Data enters through forwarders, scripted input, file input, and so forth according to the settings and deployment. The data is forwarded to the indexer, or indexers/search heads can be placed together.
- **Indexing/searching:** Indexers in this type of enterprise deployment receive data from forwarders. They store and index incoming data. The indexer in a small enterprise deployment also behaves as a search head. It searches a user query and performs a local search on data from the data input phase.

Now, let's look at medium-scale enterprise deployment.

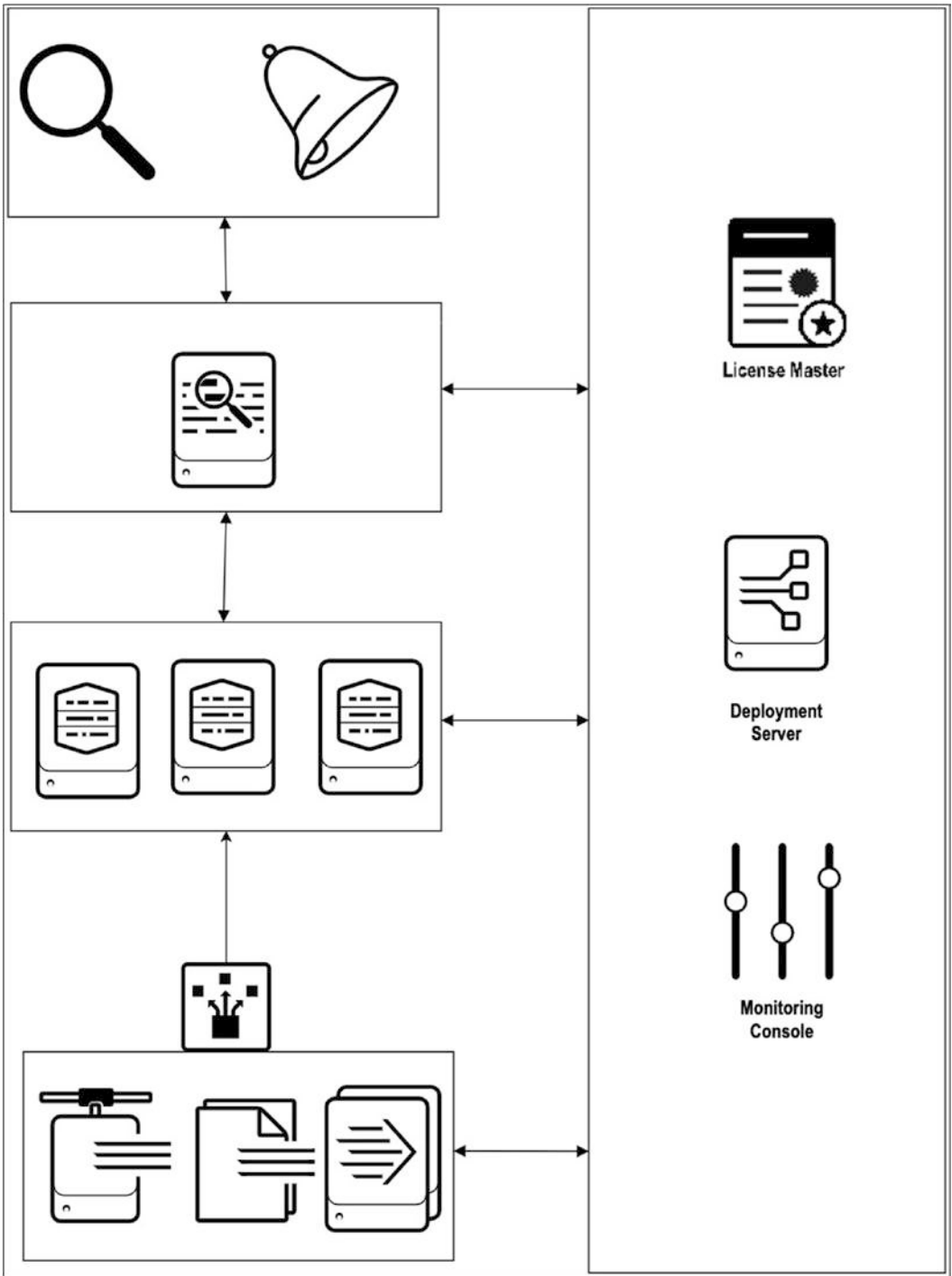


*Figure 13-7. Small Enterprise Splunk Constituent Architecture Deployment*

## Medium-Scale Enterprise Deployment

If the indexing volume ranges between 600 and 1000 gb/day and the users in an organization are between 10 and 100, then medium scale Enterprise Deployment are favourable. It has various forwarders forwarding the data to the indexer. In these types of enterprise, the search head are placed on top of the indexer to make the service run more efficiently.

Figure 13-8 shows a medium enterprise deployment.



**Figure 13-8.** Medium Enterprise Splunk Constituent Architecture Deployment

- **Data input:** The data enters through forwarders, scripted inputs, file input, and so forth according to the required settings and deployment. The data is forwarded to the indexer or an indexer/search head is placed together.
- **Indexing:** Indexers in this type of enterprise deployment receive data from the forwarders. They store and index the incoming data.
- **Searching:** In this type of enterprise, separate search heads are configured to handle the users' query request.

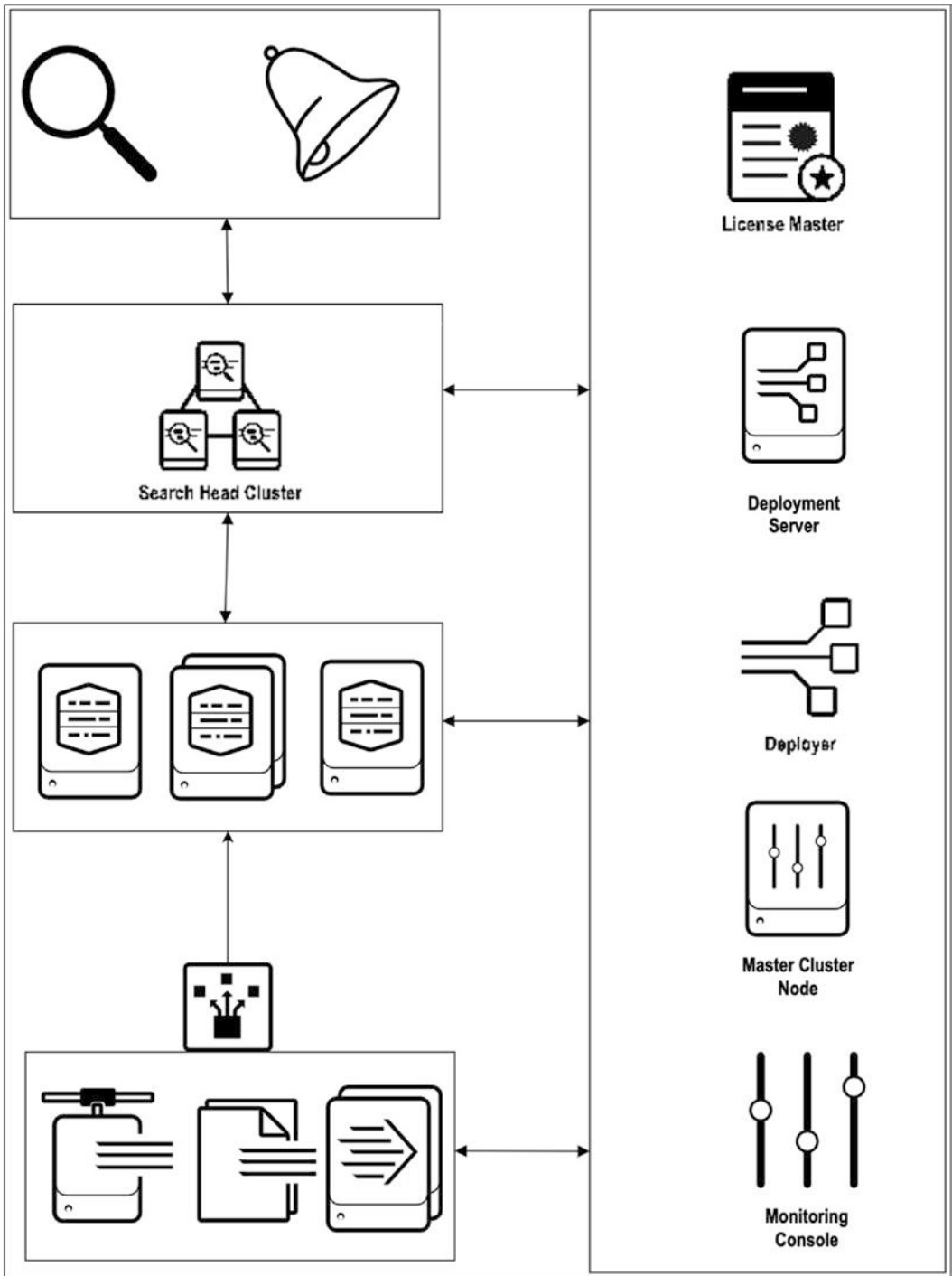
If the incoming data suddenly increases in a medium-scale enterprise, you need to make it more scalable. To do so, you need to add more indexers and search heads.

Let's now look at a large-scale enterprise deployment.

## Large-Scale Enterprise Deployment

If an organization has an indexing volume of more than 1 TB/day, and there are more than 300 users, a large-scale enterprise deployment is required. A large-scale enterprise deployment has various forwarders forwarding data to the indexer and indexer search heads are placed on top of that. Search head captains are also present so that searching is done efficiently.

Figure 13-9 shows a large enterprise deployment.



*Figure 13-9. Large Enterprise Splunk Constituent Architecture Deployment*



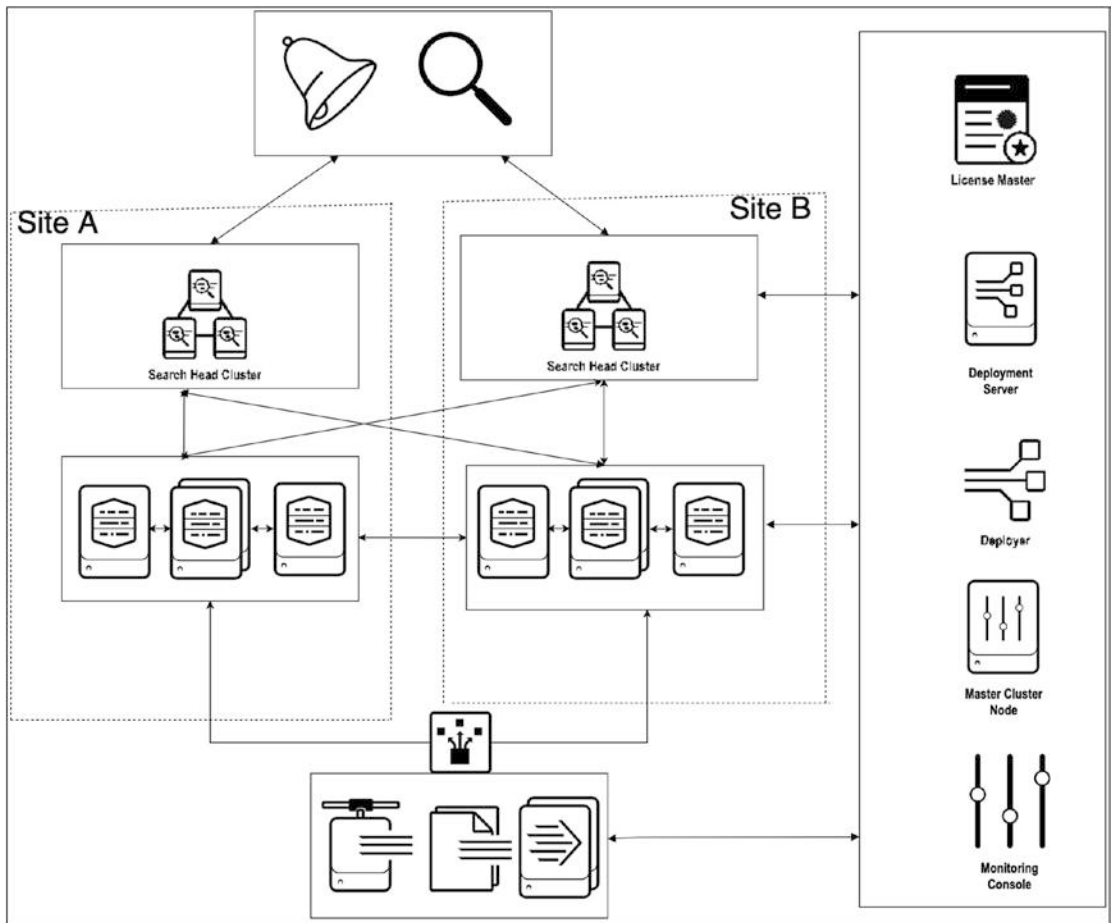
- **Data input:** Data enters through forwarders, scripted input, file input, and so forth according to the settings and deployment. Load-balancing capabilities are deployed to make data spread across the series of indexers.
- **Indexers:** In this type of deployment, indexers receive data from forwarders to store and index.
- **Search head:** In this type of enterprise, a greater number of search heads are configured to handle large amounts of query requests from users.

By now, you know that Splunk Enterprise can handle large amounts of incoming data, but from a single-site deployment. However, since all components are placed in one site, it raises security concerns. What if the single site is damaged in a disaster, like a flood or an earthquake? To avoid this situation, Splunk supports multisite clustering.

## Multi-site (M3/M13)

Multi-site (M3/M13) uses Search Head Cluster for horizontal scaling and removes single point failure from the search tier in each site. It has Search Head Cluster (SHC) per site to increase available search capacity, distribute scheduled work load and optimal user failover. Multi-site (M3/M13) also include implementation of distributed cluster deployment useful to replicate data among multiple sites by configuring site replication and search factor.

Multisite indexer clustering is depicted in [Figure 13-10](#).



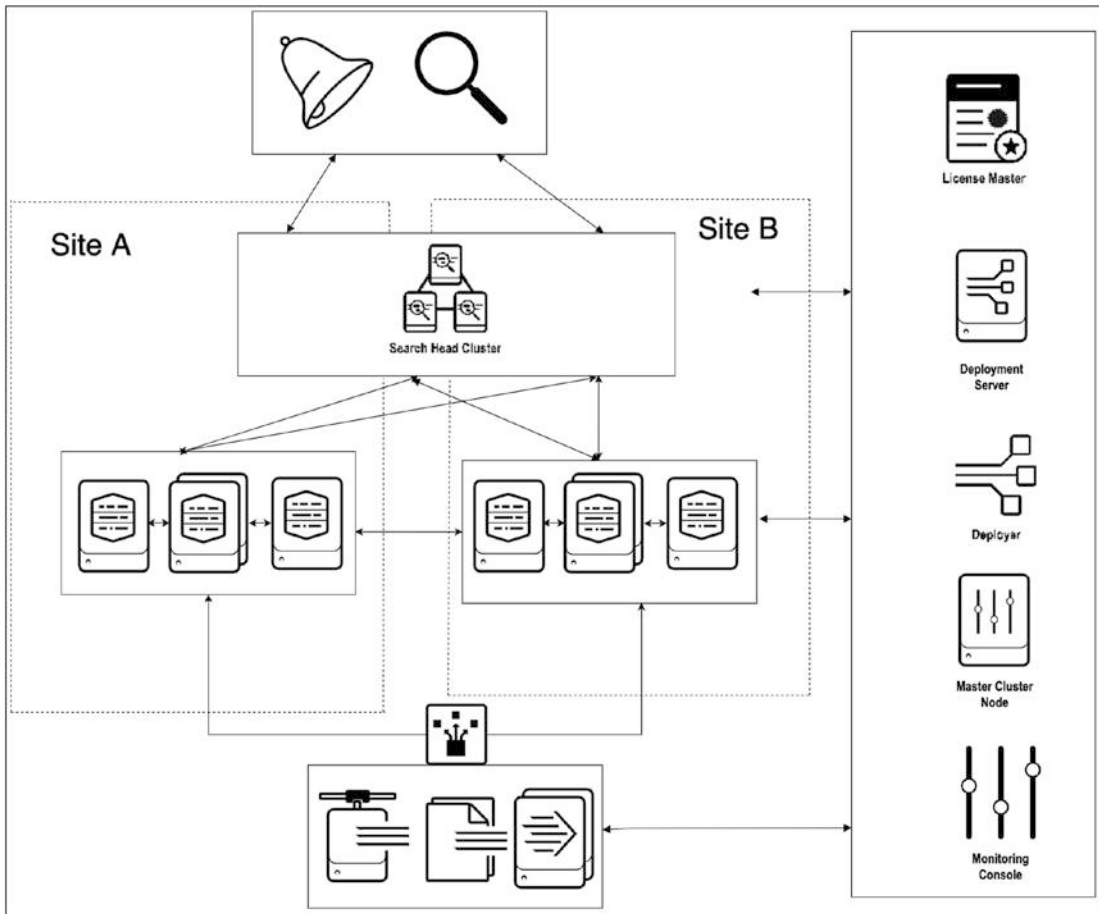
**Figure 13-10.** Multi-site (M3/M13) Constituent Architecture Deployment

- **Data input:** Data enters through forwarders, scripted input, file input, and so forth according to the settings and . Load-balancing capabilities are deployed to make the data spread across the series of indexers.
- **deployment.** Load-balancing capabilities are deployed to make the data spread across the series of indexers.
- **Indexers:** In this type of deployment, indexers receive data from forwarders to store and index the incoming data. The indexer also has a master node that regulates the function of the indexer cluster and coordinates how buckets are replicated among different sites.

- **Search head clustering:** Search head clusters are configured for individual sites. Each site has its own search head clustering.

## Multi-site (M4/M14)

Multi-site (M3/M13) and Multi-site (M4/M14) are almost similar. The only changes is it includes one stretched search head that manages two sites. It results in optimal failover for users in case of a search node or data center failure.



**Figure 13-11.** Multi-site (M4/M14) Constituent Architecture Deployment

- **Data input:** Data enters through forwarders, scripted input, file input, and so forth according to the settings and deployment: In this type of deployment, indexers receive data from forwarders to store and index the incoming data. The indexer also has a master node that regulates the function of the indexer cluster and coordinates how buckets are replicated among different sites.
- **Search head clustering:** Search head clusters are configured according to your enterprise design. Search head clustering is stretched among shared sites.
- **Indexers:** In this type of deployment, indexers receive data from forwarders to store and index the incoming data. The indexer also has a master node that regulates the function of the indexer cluster and coordinates how buckets are replicated among different sites.

## Splunk Architecture Practices

This section offers a peek at the Splunk architecture roles and the practices. There is one problem statement. A company named XYZ wants to move to Splunk. XYZ has provided its requirements, so let's use them to apply best practices in migrating existing solutions to Splunk.

### Use Case: Company XYZ

Your client, Company XYZ, wants to move to Splunk. It instituted its current infrastructure to meet PCI compliance ecommerce requirements. The logging system implemented to address those requirements is now at overflow. XYZ is running out of resources to store logs, which is affecting their ability to monitor their operational, security, and compliance logs. As a result, XYZ wants to move to Splunk. Its current environment uses Linux and Windows-based logs. It directly correlates with syslog.

The current data sources are firewall services and ecommerce, proxy, database, and network logs. The retention period differs among the different data sources. For firewalls, the retention period is 90 days. For app services, the retention time is 120 days. For ecommerce, the retention time is 365 days. For proxy data, the retention is 90 days. For

databases, the retention is 180 days. For networks, the retention is 30 days. App services, networks, and proxies are Windows and Linux-based. The others are Linux only.

PCI compliance is only for ecommerce data; the others do not have a compliance-based data feed. The level of access is defined in four levels: security, Ops, sales, and support. Security has access to all logs. The Ops team has access to the app services, ecommerce, database, and network logs. The support team has access to app services and ecommerce logs. The sales team has access to only the ecommerce logs.

The raw data compression rate is around 15% (provided), and the compression index rate is around 35% (provided). The average amount for a firewall data source is 40 GB/day. App services have 20 GB/day, ecommerce has 102 GB/day, a proxy data source has 30 GB/day, databases have 40 GB/day, and the network has 10 GB/day.

After discussions with the client, you understand how you can monitor the firewall, ecommerce, proxy, and database data sources. For app services, you need to use scripted input. To capture network logs, you need to use network ports.

Company XYZ has 38 users. It has 9 users in the security group; they are active from 12 to 18 hours/day. It has 12 users in the sales group; they are active from 7 to 11 hours/day. It has 10 users in the Ops group; they are active 6 to 13 hours/day. It has 9 users in the support group; they are active 2 to 8 hours a day.

The company needs 18 dashboards, 75 alerts, 85 reports, and 5 data models. The data rolling period from a hot bucket to a cold bucket is 30 days.

The Splunk architecture designed to have six stages.

- **Splunk data input** is the initial stage of data gathering. Solution architects had a series of calls with company XYZ to gather requirements such as daily incoming data, data sources, retention periods, and access visibility access to each group.
- **Splunk index calculation** is the second stage. It is based on the raw data compression ratio, retention days, and the index file compression ratio. You calculate the base size of raw data and base size of index calculation, which helps you design the Splunk index and get an overview of the current architecture.
- **Splunk total saved data** is the third stage. In this stage, you calculate the total required disk size based on the raw data compression ratio, retention days, index file compression ratio, replication factor, and search factor. Total saved data provides insight to the Splunk architect to help him set the number of indexers needed for the infrastructure.

- **Splunk user planner** is the fourth stage. The overhead cost is calculated based on scheduled searches, concurrent users, active time of concurrent users, and index parallelization. The Splunk user planner helps the architect determine the overhead costs and plan the Splunk infrastructure design.
- **Splunk hardware and apps considerations** are the fifth stage. Disk storage, horizontal scaling, vertical scaling, index parallelization, and sizing factors for searching are investigated in this stage. Hardware and Splunk scaling considerations help Splunk architects design a scalable solution.
- **Disk size calculation** is the last stage. In this stage, you calculate the exact disk size based on all previous stages' data feed. In disk size calculation, you learn how Splunk architecture is designed.

Now let's dive into each stage.

## Splunk Data Inputs

Splunk data input is the initial stage. It comprises gathering the data sources, methods, environment, index names, retention days, visibility access, ownership, integrity, raw data compression ratio, index file compression ratio, and average daily amount of data in gigabytes. Let's have a look at each component.

- **Data Source** is the name of data input or data source.
- **Method** describes where the data is located and how to onboard data to Splunk. It can be collected through scripted input, monitor stanzas, network ports, and so forth.
- **Environment** is the operating system. It helps in planning the enterprise deployment.
- **Index Name** is the name of the index that the Splunk architect needs to plan. It helps in effective deployment.
- **Retention Days** is the number of days that data should be kept in Splunk. A retention policy depends on legal, business, and search requirements.

- **Visibility/Access** specifies which individuals or teams have access to data.
- **Ownership** defines the owner or person who controls Visibility/Access and can make notable changes in Splunk.
- **Integrity** specifies special considerations needed to maintain data integrity.
- **Compression Raw Data** is an estimated percentage of what Splunk saves by compressing raw data. For example, 15% means 100 GB of raw data can be compressed to 15 GB data.
- **Compression Index File** is an estimated ratio of what Splunk saves by compressing index file. For example, if there is 100 GB incoming raw data, and the compression index file (lexical table) is 35%, the incoming data is compressed to 35 GB data.
- **Average Data GB per Day:** The daily average amount of incoming data in gigabytes.

Figure 13-12 is a Splunk Data Input spreadsheet. Let’s have a look at it.

**SPLUNK DATA INPUT**

Data Source (Name)	Method (Name)	Environment (linux&Windows)	Index Name (Index name)	Retention Days (In Days)	Visibility/Access (Group Name)	Ownership (Group Name)	Integrity (Yes/No)	Compression Raw Data (In %)	Compression Index File (In %)	Average Data GB Per Day (in Gigabit)
Firewall	Monitor	Linux	firewall	90	Security	Security	No	15%	35%	40
App Services	Scripted Input	Windows,linux	app	120	Security,Ops,Support	Security	No	15%	35%	20
Ecommerce	Monitor	linux	ecom	365	security ,Ops,Support,Sales	Security	Yes	15%	35%	102
Proxy	Monitor	linux,windows	proxy	90	Security	Security	No	15%	35%	30
Database	Monitor	linux	db	180	Security,Ops	Security	No	15%	35%	40
Network	Network Port	Windows,linux	network	30	Security,Ops	Security	No	15%	35%	10
										<b>242</b>

*Figure 13-12. Splunk Data Input*

## Splunk Index Calculation

Splunk index calculation is the second stage. Let's look at each component.

- **Compression Raw Data** is an estimated ratio of what Splunk saves by compressing raw data. **Compression Raw Data = compression percentage \* Average Data per Day**. For example, the firewall data source has 40 GB of average data per day, and the compression percentage is 15% of raw data. So, raw data compression for a firewall source type is 6 GB/day.
- **Compression Index File** is an estimated ratio what Splunk saves by compressing index file. **Compression index File = compression index file percentage \* Average Data per Day**. For example, the firewall data source has 40 GB of average data per day, and the compression percentage is 35% of the index file. So, the compression index file for a firewall source type is 14 GB/day.
- **Base Size of Raw Data:** The raw data's base size constitutes the total size of raw data per data source. To calculate, **Base Size of Raw = Compression Raw Data \* Retention Days**. For example, the firewall data source has 6 GB/day for raw data compression and retention is 90 days. So, the base size of raw data for the firewall source type is 540 GB.
- **Base Size of index File:** The base size of raw data constitutes the total raw data per index file. To calculate, **Base Size of index=Compression index File \* Retention Days**. For example, the firewall data source has 14 GB/day for data index and retention is 90 days. So, the base size of index data for the firewall source type is 1260 GB.

Figure 13-13 is a Splunk Index Calculation spreadsheet. Let's have a look at it.

SPLUNK INDEX CALCULATION							
Data Source (Name)	Retention Days (In Days)	Visibility/Access (Group Name)	Average Data GB Per Day (In Gigabit)	Compression Raw Data (In 15 %)	Compression Index File (In 35 %)	Base Size Of Raw Data	Base Size Of Index file
Firewall	90	Security	40	6	14	540	1260
App Services	120	Security,Ops,Support	20	3	7	360	840
Ecommerce	365	Security,Ops,Support,Sales	102	15.3	35.7	5584.5	13030.5
Proxy	90	Security	30	4.5	10.5	405	945
Database	180	Security,Ops	40	6	14	1080	2520
Network	30	Security,Ops	10	1.5	3.5	45	105
			242	36.3	84.7	8014.5	18700.5

Figure 13-13. Splunk Index Calculation



## Splunk Total Disk Size

Splunk Total Disk Size is the third stage. Let’s look at each component.

- Search Factor** specifies the number of searchable inside index clustering. Searchable buckets have both raw data and index files. When you configure a master node in Splunk Enterprise, you set a search factor that defines the number of searchable copies per bucket that must be maintained in the entire index cluster. The default value of the search factor is 2.
- Replication Factor** specifies the total number of raw data copies the cluster should maintain. Indexers store incoming data in buckets, and the cluster maintains copies of each bucket across the environment. Generally, the replication factor value is one more than the search factor.
- Total Disk Size** comprises the base size of raw data, the base size of the index file, the search factor, and the replication factor. To calculate, Total Disk Size = Base Size of Raw Data \* Replication Factor + Base Size of Index File \* Search Factor. For example, when you want to calculate total disk size, you consider the firewall data source has a Base Size of Raw Data = 540 GB, Base Size of Index File = 1260 GB, Search Factor = 2 and Replication Factor = 3. You get Total Disk Size = 4140 GB.

Figure 13-14 is a Splunk Total Disk Size spreadsheet based on information in the use case. Let’s have a look at it.

<i>Splunk Total Disk Size</i>						
Data Source	Retention Days	Base Size Of Raw Data	Base Size Of Index file	Search Factor	Replication Factor	Total Disk Size
(Name)	(In Days)	(In GB)	(In GB)	(Count)	(Count)	(In GB)
Firewall	90	540	1260	2	3	4140
App Services	120	360	840	2	3	2760
Ecommerce	365	5584.5	13030.5	2	3	42814.5
Proxy	90	405	945	2	3	3105
Database	180	1080	2520	2	3	8280
Network	30	45	105	2	3	345
		8014.5	18700.5			61444.5

*Figure 13-14. Splunk Total Disk Size*

## Splunk User Planner

The Splunk User Planner is the fourth stage. Let's look at each component.

- **Ad hoc searches** are also called unscheduled searches. They can be run in several ways. For example, when you are invoking a search command using the Splunk search processing language, it invokes ad hoc searches.
- **Scheduled searches** are scheduled using the search scheduler or created as part of a report acceleration or data model acceleration. The report, dashboard, and alert are the **searches** are generated for report acceleration or data model acceleration. Summarization reports are also a part of summarization searches.
- **Index parallelization** allows the indexer to maintain multiple pipeline sets. It allows the indexer to create and manage multiple pipelines, provide more cores to process it, and increase the limit of indexer I/O capacity.
- **Real-time searches** refresh in real time, continually pushing old data and acquiring new data, depending on the bounds.
- **Concurrent users** provide a CPU core for every search that the user invokes. If multiple users are logged in and running the searches, all the available CPU cores are quickly exhausted. For example, consider four cores and 12 users who are running different searches. The time to process search increases because the number of available cores is less than the number of requests.

Figure 13-15 is a Splunk User Planner spreadsheet based on information in the use case. Let's have a look at it.

<b>Splunk user Planner</b>		
<b>Parameter</b>	<b>Source Name</b>	<b>Count</b>
<b>Scheduled Searches</b>	Dashboard	18 dashboard
	Alert	75 alert
	Reports	85 reports
	Accelerate Data Model	5 models
<b>Concurrent Users</b>	Security	7 users
	Sales	12 users
	Ops	10 users
	Support	9 users
<b>index Parallelization</b>	-	2 (Parallel Ingestion Pipeline)
<b>Real Time Searches</b>	Security	12-18 hours
	Sales	7-11 hours
	Ops	6-13 hours
	Support	2-8 hours

Figure 13-15. Splunk User Planner

## Hardware and Splunk Scaling Considerations

Hardware and Splunk scaling is the fifth stage. Let’s look at each component.

- **Disk storage** comprises input/output operations per second. It helps measure disk storage. Hard drives read and write at different speeds. For bucket storage, Splunk uses a storage area network (SAN) and network-attached storage (NAS). SAN is for hot and warm buckets, and NAS is used for a cold bucket.
- **Horizontal scaling:** If a load is increases and you want to add more indexing power to the current Splunk architecture pool, you can use horizontal scaling to upgrade it by adding a greater number of indexers. Similarly, to provide more searching power, you can add more search heads to existing resources.

- **Vertical scaling:** If a load increases and you want to add more indexing power to the current Splunk architecture pool, you can use vertical scaling to upgrade the machine configuration by adding more CPU and cores. But there are limitations.
- **Increase index parallelization** is used only when the current system is underutilized. For example, you have 12 core indexers but only four cores are being used. In this scenario, it is recommended to increase index parallelization.
- **Sizing factors for searching** It comprises a number of scheduled searches, number of reports, total dashboards, active concurrent users, total count of searches per hour, and so forth. In the basis of all such factors Sizing factors for searching needs to be designed.

## Disk Size Calculation

Disk Size Calculation is the sixth stage of index calculation. In this stage, you design a Splunk architecture index in which you calculate the disk size.

From the information presented in the use case, you see that the ingestion volume in Splunk is 242 GB/day. The replication factor is 3, and the search factor is 2. Hot and warm buckets are saved in SAN. Cold buckets are saved NAS. According to the use case, data is in hot buckets and warm buckets for 30 days. You have four indexers with 60.5 GB volume of data indexing per day, and the overhead cost is 25%.

Figure 13-16 is a spreadsheet for disk size calculation using the information from the use case.

Data Sizing Exercise													
Replication Factor				3									
Search Factor				2									
Number of Days in Hot/Warm				30									
Data Source	GB per day	Raw Compression Rate	Index Compression Rate	Retention in Days	Base Size of Raw	Base Size of Index Files	Replicate (Y or blank)	Replicated Size on Disk	Hot and Warm	Cold	Visible to	Index	
firewall	40	0.15	0.35	90	540	1,260	Y	4,140	1,380	2,760	Security	firewall	
app	20	0.15	0.35	120	360	840	Y	2,760	690	2,070	Security.Ops.Support	app	
ecom	102	0.15	0.35	365	5,585	13,031	Y	42,817	3,520	39,297	security.Ops.Support.Sales	ecom	
proxy	30	0.15	0.35	90	405	945	Y	3,105	1,035	2,070	Security	proxy	
db	40	0.15	0.35	180	1,080	2,520	Y	8,280	1,380	6,900	Security.Ops	db	
network	10	0.15	0.35	30	45	105	Y	345	345	-	Security.Ops	network	
Totals	242				8,015	18,701		61,447	8,350	53,097			
Number of Indexers	4			60.5			Ingestion Volume per Indexer GB/Day						
Recovery space	if number of indexers lost-			1			15,362 2,088 13,274						
Overhead	25%						19,202 2,610 16,592						
Total Disk							96,011 13,048 82,963						
Disk Space per indexer							24,003 3,262 20,741						

Figure 13-16. Disk Size Exercise

Let’s quickly recap this use case and what’s needed to calculate the disk size.

- **Rep Factor:** 3
- **Search Factor:** 2
- **Number of Indexers:** 4
- **Number of Days in Hot/Warm Bucket:** 30 days
- **Overhead:** 0.25 or 25%
- **Number of Indexer Lost:** 1
- **Raw Compression Rate:** 15% or 0.15
- **Index Compression Rate:** 35% or 0.35
- **Base Size of Raw:** Compression Raw Data \* Retention Days
- **Base Size of Index Files:** Compression Index File \* Retention Days
- **Replicate Size on Disk:** Base Size of Raw Data\* Replication Factor + Base Size of Index File \* Search Factor
- **Ingestion volume per indexer:** Total GB per day/Number of Indexers
- **Hot and warm:** (Raw Compression Rate \* GB per Day \* Replication Factor) + (Index Compression Rate \* GB per Day \* Search Factor) \* Number of days in hot/warm bucket
- **Cold bucket:** (Raw Compression Rate \* GB per Day\* Replication Factor) + (Index Compression Rate \* GB per Day \* Search Factor) \* (Retention Days – Number of days in hot/warm bucket)

- **Data Storage per Indexer:** Total Replicated Size on Disk /Number of Indexer
- **Overhead Cost:** Data Storage per Indexer + Overhead \* Data Storage per Indexer
- **Total Disk Space:** Overhead Cost \* Number of Indexers + Overhead cost
- **Total Disk Space per Indexer:** Total Disk/Number of Indexer

With this, you have come to the end of this chapter. Congratulate yourself on successfully learning about capacity planning in the Splunk environment, single and multisite clustering, search heads, and designing a Splunk architecture in an enterprise deployment.

## Summary

This chapter addressed various capacity planning components and configuring search heads using Splunk Web, the CLI, and the configuration file. It also covered search head clustering, the configuration of the search head captain, multisite clustering, and the Splunk architecture. You learned about designing small-scale, medium-scale, large-scale, and multisite indexer clustering.

This chapter covered a hefty portion of the Splunk architect exam blueprint. You became familiar with the 5% of Module 2 (project requirements), 5% of Module 3 (infrastructure planning), 7% of Module 4 (resource planning), 5% of Module 18 (search head clusters), and 5% of Module 16 (multisite index cluster).

The next chapter covers the Monitoring Console, metrics.log, job inspectors, troubleshooting license violations, troubleshooting deployment issues, and troubleshooting clustering issues.

## Multiple-Choice Questions

- A. Which file configures a Splunk search peer using the configuration file?
1. server.conf
  2. deployment.conf
  3. output.conf
  4. distsearch.conf

- B. Which Splunk component performs indexing and responds to search requests from the search head?
  - 1. forwarder
  - 2. search peer
  - 3. license master
  - 4. search head cluster
  
- C. Which of the following is a valid distributed search group?
  - 1. [distributedSearch:test] default=false servers=server1,server2
  - 2. [searchGroup:test] default=false servers=server1,server2
  - 3. [distributedSearch:test] default=false servers=server1:8089,server2:8089
  - 4. [searchGroup:test] default=false servers=server1:8089,server2:8089
  
- D. Which Splunk component consolidates individual results and prepares reports in a distributed environment?
  - 1. indexers
  - 2. forwarder
  - 3. search head
  - 4. search peers
  
- E. What hardware attribute needs to be changed to increase the number of simultaneous searches (ad hoc and scheduled) on a single search head?
  - 1. disk
  - 2. CPUs
  - 3. memory
  - 4. network interface cards

- F. Which Splunk component does a search head primarily communicate with?
1. indexer
  2. forwarder
  3. cluster master
  4. deployment server

**Answers**

- A. 4  
B. 2  
C. 3  
D. 3  
E. 2  
F. 1

## References

- <https://www.splunk.com/pdfs/technical-briefs/splunk-validated-architectures.pdf>
- Splunk 7.x Quick Start Guide: Gain Business Data Insights from Operational Intelligence
- <https://docs.splunk.com/Documentation/Splunk/8.1.1/InheritedDeployment/Deploymenttopology>
- <https://docs.splunk.com/Documentation/Splunk/8.1.1/InheritedDeployment/Diagramyourdeployment>



## CHAPTER 14

# Troubleshooting in Splunk

This chapter focuses on analyzing activity and diagnosing problems within Splunk Enterprise. You learn about the Monitoring Console used for troubleshooting and all its components. You get an overview of all the log files useful for troubleshooting. And you learn about the job inspector, what happens when a license violation occurs, and troubleshooting deployment and clustering issues.

This chapter covers the following topics.

- The Monitoring Console
- The log files for troubleshooting
- The metrics.log
- Job inspectors
- Troubleshooting license violations
- Troubleshooting deployment issues
- Troubleshooting clustering issues

By the time you complete this chapter, you will have learned 20% of the Splunk architect exam blueprint.

## Monitoring Console

The Monitoring Console in Splunk consists of a set of reports, dashboards, alerts, and health checks designed to provide a detailed view and information about your Splunk Enterprise performance. The Monitoring Console is available only to admin users. The Monitoring Console is one of the biggest troubleshooting assets. The following are a few of its most important functions.

- Single instance deployment
- Multi-instance deployment

- Monitors the system’s health
- Configures forwarder monitoring

## Single Instance Deployment Monitoring Console

The single instance deployment Monitoring Console is used for a single instance where the indexer, license master, KV Store, and search head are in one instance. To configure Splunk deployment for a single instance, refer to the following steps.

1. In Splunk Web, go to the Settings page.
2. Go to Monitoring Console.
3. Go to the Monitoring Console’s Settings page and then go to General Setup.
4. Click Standalone.
5. Select the indexer, license master, KV Store, and search head under Server Roles (see Figure 14-1).



Figure 14-1. Monitoring Console:Single Instance

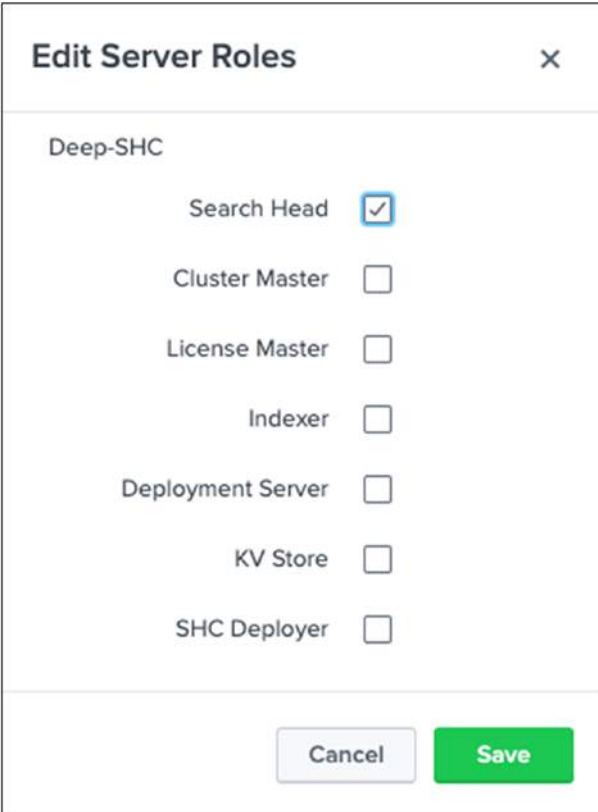
## Multi-Instance Deployment Monitoring Console

The multi-instance deployment Monitoring Console is used for distributed environments where the indexer, license master, KV Store, and search head are not located within a single instance. To configure Splunk deployment for a multi-instance, refer to the following steps.

1. In Splunk Web, go to the Settings page.
2. Go to the Monitoring Console.

3. Go to the Monitoring Console's Settings page and then go to GENERAL SETUP.
4. Turn on Distributed Mode.
5. Assign the roles to the node and configure the deployment.
6. Click Save.

Figure 14-2 shows how you edit the server roles.



**Edit Server Roles** ×

Deep-SHC

- Search Head
- Cluster Master
- License Master
- Indexer
- Deployment Server
- KV Store
- SHC Deployer

Cancel Save

**Figure 14-2.** *Monitoring Console:Server Role*

The following explains how to reset the server role after a restart.

1. In the Monitoring Console, click Settings and go to General Setup.
2. Click Distributed Mode.
3. To edit server roles, go to Remote Instance, and click Edit.

4. Select or edit the server role.
5. Click Apply changes.

Figure 14-3 shows how to edit the server role and assign roles to the instance.

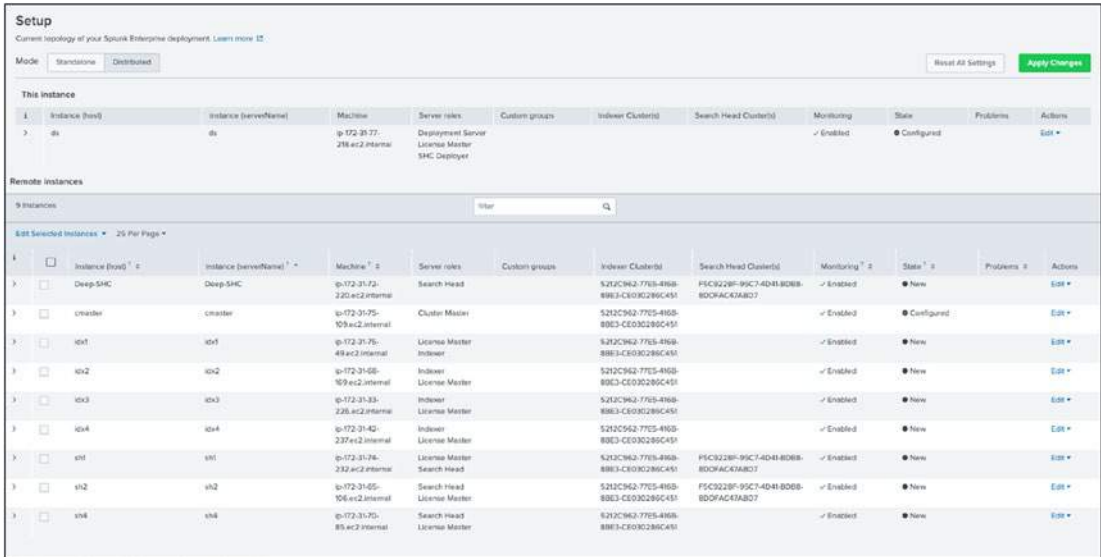


Figure 14-3. Monitoring Console Server Role:Distributed mode

6. Once you have made all changes in the server role, click Save and continue to the Overview page.
7. Click Go to Overview (see Figure 14-4).

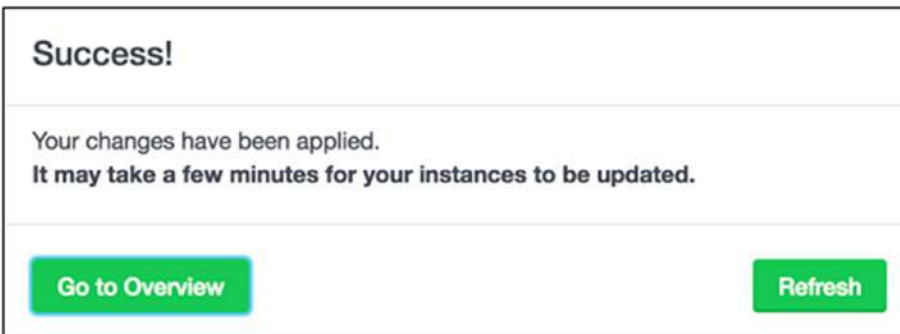
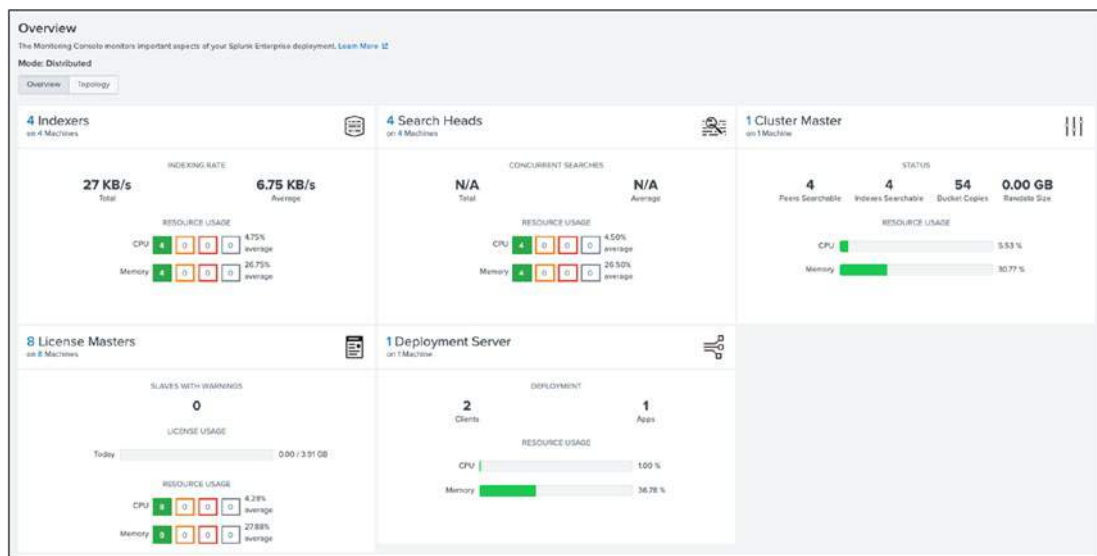


Figure 14-4.

Figure 14-5 shows the multi-instance deployment Monitoring Console. There are four indexers, four search heads, and one cluster master.



*Figure 14-5. Monitoring Console:Deployment*

## Monitor System Health with Monitoring Console

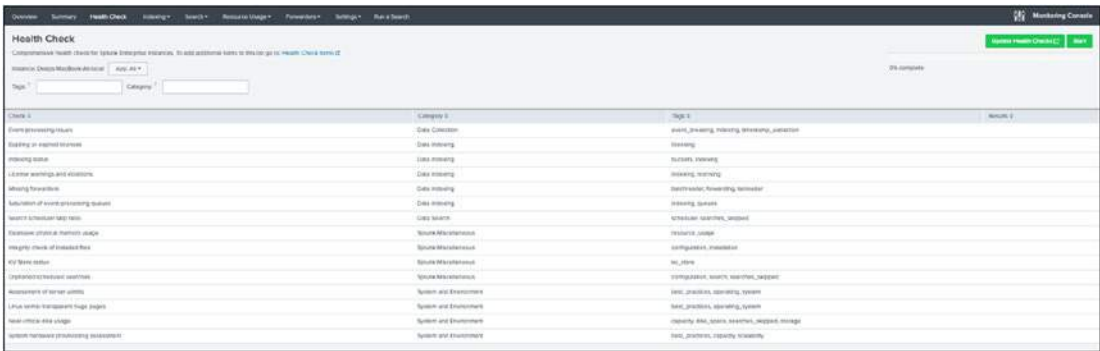
The health check monitors overall system health. If a particular forwarder is decommissioned, it remains on the forwarder table until you remove it. The forwarder asset table removes decommissioned nodes from the system. Platform alerts are saved searches in Splunk. Platform alerts notify the Splunk administrator. If any condition is triggered using the Monitoring Console, you know which alerts have been configured in the system. There are a few preconfigured alerts that come with the health check. You can modify an existing health check or create a new one.

Do the following to run a health check in the system.

1. In Splunk Web, go to the Settings page.
2. Go to Monitoring Console.
3. Go to the Monitoring Console's Settings page and then go to General Setup.

4. Go to Health check.
5. Start the health check.

Figure 14-6 shows a health check in the system. If you encounter an error, you need to start troubleshooting it. The health check helps you understand the current status of your system.

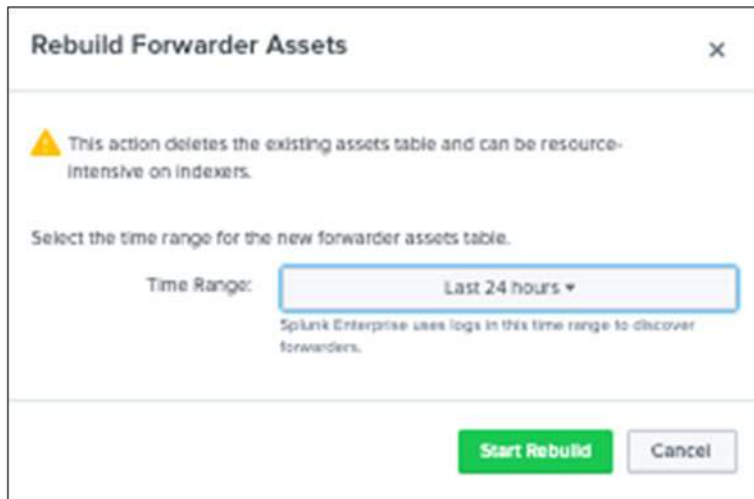


**Figure 14-6.** Monitor Console: Health Check

Do the following to rebuild a forwarder asset table.

1. In the Monitoring Console, click Settings.
2. Go to Forwarder Monitoring Setup.
3. Click Enable Forwarding if it's disabled.
4. Click Save.
5. Click Rebuild Forwarder Asset Table.
6. Select Timeline.
7. Click Start Rebuild.

I selected the **Last 24 hours** time range to rebuild the forwarder asset table (see Figure 14-7).



**Figure 14-7.** Monitor Console:Rebuild Forwarder Asset Table

The Monitoring Console removes missing forwarders by rebuilding the forwarder asset table.

## Configure Forwarder Monitoring for the Monitoring Console

The Monitoring Console monitors incoming data from forwarder connections. The measurement of incoming data by a forwarder is done using the **metrics.log**.

The following explains how to set up a forwarder instance view.

1. In Splunk web, go to the Settings page.
2. In the Monitoring Console, click Forwarders.
3. Go to Forwarder Instance.
4. Select the instance name and select a time range.

Figure 14-8 shows a forwarder instance where the instance is dell-pc and time range is **5 minute window**.

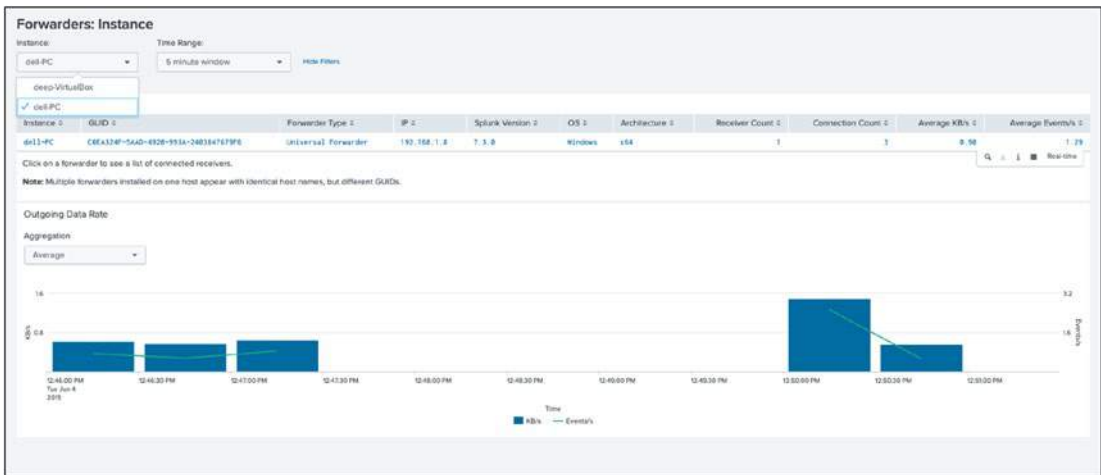


Figure 14-8. Monitoring Console:Forwarder Monitoring

Next, let’s look at log files for troubleshooting and metric analysis.

## Log Files for Troubleshooting

The log files in the Splunk are located at \$SPLUNK\_HOME/var/log/splunk/ file name. These log files are used in troubleshooting Splunk Enterprise. The internal logs help you troubleshoot or are used for metric analysis. Table 14-1 covers all the major log files needed for troubleshooting. In this chapter we will cover metrics.log file in detail and to study more about log files for troubleshooting go to <https://docs.splunk.com/Documentation/Splunk/7.3.0/Troubleshooting/WhatSplunklogsaboutitself>.



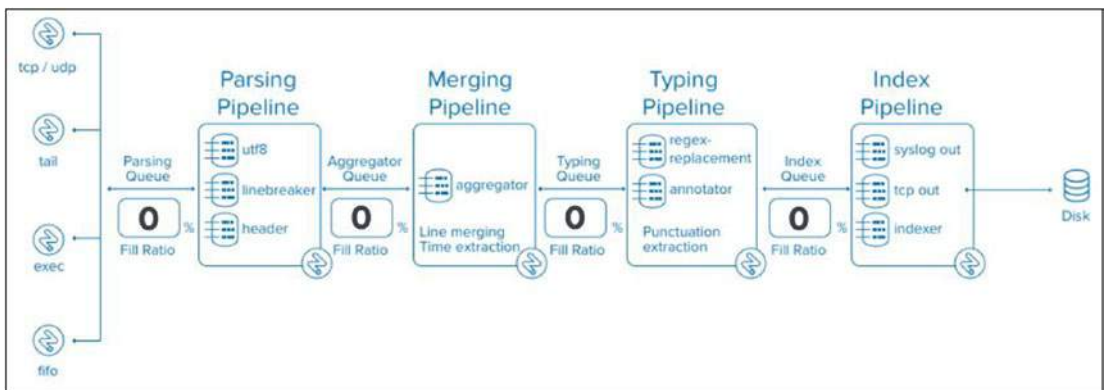
**Table 14-1.** Internal Log Files for Analysis & Troubleshooting

Attribute	Value
audit.log	This file consists of information about user activity and modifying knowledge objects, and saved searches in Splunk.
btool.log	The Btool is used for troubleshooting Splunk configuration files. You use btool.log to log Btool activity.
metrics.log	This file consists of performance and system data, information about CPU usage by internal processors, and queue usage in Splunk's data processing.
splunkd.log	This file is used for troubleshooting. Any stderr messages generated by scripted input, scripted search command, and so on, are logged here.
splunkd_access.log	This file is from splunkd through the logged UI.

Next, let's discuss the metrics.log file and the important group associated with it.

## The metrics.log File

The metrics.log file reveals a picture of the entire Splunk Enterprise. Data in Splunk moves through a data pipeline in phases. These phases include the input pipeline, parsing pipeline, merging pipeline, typing pipeline, and index pipeline, as shown in Figure 14-9.

**Figure 14-9.** Splunk Enterprise Data Pipeline

- The **input pipeline** mainly consists of data sources. The host, source, and source type are added in this stage. This pipeline is the first phase. It includes the TCP/UDP pipeline, tailing, FIFO pipeline, and exec pipeline methods. `inputs.conf`, `wmi.conf`, and `props.conf` are the configuration files in the input pipeline.
- The **parsing pipeline** is the second phase. This pipeline has two processors: **reader in** and **sendout**. `props.conf` file is the file of interest in the parsing pipeline. The reader in processor takes data from the parsing queue, and sendout sends data to the aggregator queue. Parsing UTF-8, line breakers, and headers are included in the parsing phase.
- The **merging pipeline** is the third phase. This pipeline has two processors: **aggregator** and **winparsingqueue**. The aggregator processor identifies the events boundary and does timestamp extractions, and the winparsingqueue processor has its aggregator and feeds data to a typing queue. Line merging for multi-line events and time extraction for each event is done in this phase. `props.conf` is the configuration file included in the merging pipeline. `SHOULD_LINE_MERGE`, `BREAK_ONLY_BEFORE`, and `MUST_BREAK_AFTER`, `MAX_EVENTS` are included in the merging pipeline.
- The **typing pipeline** is the fourth phase. It has two processors: `readerin` and `sendout`. The `props.conf` file and `transforms.conf` are the file of interest in the typing pipeline. The `readerin` processor read data from aggregate queue and `sendout` sends data to index queue. `Transforms -xxx`, `SEDCMD`, `SOURCE_KEY`, `DEST_KEY`, `REGEX`, and `FORMAT` are also included in the typing pipeline.
- The **index pipeline** is the fifth phase. This pipeline has `indexin` processor. `indexes.conf` and `outputs.conf` is the file of interest in the index pipeline. The `indexin` processor is responsible to read data from index queue. `Tcp-output-generic-processor`, `syslog-output-generic-processor`, `fishbucket`, `signing`, `indexer`, and `index_thruput`, are included in the index pipeline.

Consider there is a data input which is surging events. It is hidden in data and it could be difficult to sort it out. So in such scenario, one place to look is the Splunk internal `metrics.log`. The `metrics.log` file in Splunk is a report that takes input every 30 seconds. It comprises introspection information useful for reviewing product behaviour.

To troubleshoot metrics.log file in splunk web use internal index or you can use the debug mode in Splunk to look at the metrics.log located at \$Splunk\_home/var/log/splunk/metrics.log. The metrics.log file consists of structure of lines classified by group parameter which helps in defining the type of metrics data. There are many groups in the metrics.log file; the following are a few of the important groups.

- pipeline messages
- queue messages
- thruput messages
- tcpout connections messages
- udpin\_connections messages
- bucket\_metrics messages

For more information on metrics.log, refer to <https://docs.splunk.com/Documentation/Splunk/8.0.0/Troubleshooting/metricslog>.

---

Metrics data is 150 bytes per event in an Enterprise license.

---

## Pipeline Messages

Pipeline messages consist of detailed reports on the Splunk pipeline. They are placed together to process and manipulate events flowing in and out of the Splunk system. A pipeline message tracks the time used by each CPU and understands process execution.

The structure of a pipeline message is determined in the following event.

12/1/19

```
5:17:10.335 PM12-01-2019 17:17:10.335 +0530 INFO Metrics - group=pipeline,
name=typing, processor=metricschema, cpu_seconds=0, executes=405,
cumulative_hits=8314
```

- **Timestamp:** The event consists of a timestamp that logs when the event occurred in the system.
- **group=pipeline:** This reflects and catches the CPU sendout execution time. Plotting total CPU seconds by processor shows you where the CPU time is going in indexing activity.

## Queue Messages

Queue messages in Splunk consist of aggregate across events and can tell you where the indexing bottlenecks are located.

The structure of a queue message is determined in the following event.

12/1/19

8:25:19.839 PM

```
12/01/2019 20:25:19.839 +0530 INFO Metrics - group=queue, blocked=true, name=typingqueue, max_size_kb=500, current_size_kb=432, current_size=955, largest_size=966, smallest_size=966
```

- **Timestamp:** The event consists of a timestamp that logs when the event occurred in the system.
- **group=queue:** Queue data consist of aggregate that reveal bottlenecks in the network. `current_size` is an important value as it reflects the amount of data inserted in the indexer at that time. For example:-Currently, `current_size =955`, which reflects the amount of data inserted in the indexer is more at that time which would lead to bottleneck and ideally `current_size` should be between 0 and 500 which means indexing system is working fine.

## Thruput Messages

The thrupt messages in Splunk measure the rate at which raw events are processed in the indexing pipeline. Throughput numbers are the size of the “raw” the events traversing through the system.

The structure of a message is determined by the following event.

12/1/19

11:10:52.469 PM

```
12-01-2019 23:10:52.479 +0530 INFO Metrics - group=thruput, name=thruput, instantaneous_kbps=3.3666908659618975, instantaneous_eps=16.99733361182728, average_kbps=2.1346675475642374, total_k_processed=46162, kb=104.3837890625, ev=527, load_average=1.48828125
```

- **Timestamp:** The event consists of a timestamp that logs when an event occurred in the system.
- **group=thruput:** This measures the rate at which raw events are processed. It is best for insights during tuning performance or evaluating an average catchall indexing since splunk enterprise is started. Thruput helps determine which data categories are busy and help on performance tuning or evaluating indexing load.

## Tcpout Connection Messages

tcpout connection messages in Splunk provide information about the system; specifically, that the tcpout connection is connected to an open socket.

The structure of a tcpout connection message is determined in the following event.

12/1/19

11:10:52.469 PM

```
12-01-2019 23:10:52.479 +0530 INFO Metrics - group=tcpout_connections,
name=undiag_indexers:191.11.41.67:9997:0, sourcePort=9009, destIp=191.11.41.67
destPort=9997, _tcp_Bps=28339066.17, _tcp_KBps=21669.07, _tcp_avg_
thruput=29532.96, _tcp_Kprocessed=808751, _tcp_eps=31211.10, kb=807353.91
```

- **Timestamp:** The event consists of a timestamp that logs when the event occurred in the system.
- **group=tcpout\_connections:** tcpout\_connections measure the number of bytes that Splunk has successfully written to the socket. tcp\_avg\_thruput demonstrates the average rate of bytes flowing in the Splunk system since it started.

## udpin\_connections Messages

udpin\_connections messages in Splunk are essentially metering the UDP input.

The structure of `udpin_connections` messages is determined by the following event.

12/1/19

11:10:52.469 PM

```
12-01-2019 23:10:52.479 +0530 INFO Metrics -group=udpin_connections, 8001,
sourcePort=8001, _udp_bps=0.00, _udp_kbps=0.00, _udp_avg_thruput=0.00, _
udp_kprocessed=0.00, _udp_eps=0.00
```

- **Timestamp:** The event consists of a timestamp that logs when the event occurred in the system.
- **group=udpin\_connections:** `udpin_connection` measures the bytes that Splunk has written to the UDP input. `udp_avg_thruput` demonstrates the average rate of bytes flowing in the UDP input since it started. It helps you understand the flow of data in your current system.

## bucket\_metrics Messages

`bucket_metrics` messages in Splunk are a detailed summary report on the status of the bucket. You can fetch the total count in hot buckets, cold buckets, and frozen buckets.

The structure of `bucket_metrics` message is determined in the following event.

12/2/19

06:34:21.543 AM

```
12-02-2019 06:34:21.543 +0530 INFO Metrics - group=bucket_metrics, current_
hot=6, current_hot_replicas=0, current_total=68, created=0, created_
replicas=0, rolled=0, frozen=0, total_removed=0
```

- **Timestamp:** The event consists of a timestamp that logs when the event occurred in the system.
- **group=bucket\_metrics:** This shows the organization's bucket status in Splunk. `current_hot` reflects the total number of hot buckets in your organization. `current_total` displays the total number of buckets in your entire organization.

Next, let's look at Splunk search job inspectors to troubleshoot search performance.

## Job Inspector

The Splunk search job inspector troubleshoots search performance and knowledge objects. The search job inspector provides a detailed summary report on Splunk's search performance.

How does a job inspector help you troubleshoot?

The key information that is displayed by a search job inspector is execution cost and search job properties that play a major role in troubleshooting.

- **Execution cost** displays information about the components of the search and the amount of impact each component has on the search's output.
- **Search job** properties list the characteristics of the search. Each command in the search job has a part and parameter. The value of the parameter reflects the time spent on each command.
- **EPS (Event Per Second)**= Search Job properties derive the events per second so that you can understand the system's processing power and pipelines. (EPS=total events/time).

## Job Inspector Example Query

In this section, you run an example query to get group throughput events from the internal index and troubleshoot the query using a job inspector.

```
index="_internal" "group=throughput"
```

An execution cost query is shown in [Figure 14-10](#).

This search has completed and has returned **5,021** results by scanning **33,143** events in **1.07** seconds.  
 (SID: 1575302020.677) [search.log](#)

Execution costs

Duration (seconds)	Component	Invocations	Input count	Output count
0.00	command.fields	17	5,021	5,021
0.78	command.search	17	-	5,021
0.02	command.search.expand_search	2	-	-
0.02	command.search.calcfields	16	33,143	33,143
0.00	command.search.expand_search.calcfield	2	-	-
0.00	command.search.expand_search.fieldallaser	2	-	-
0.00	command.search.expand_search.kv	2	-	-
0.00	command.search.expand_search.lookup	2	-	-
0.00	command.search.expand_search.sourcetype	2	-	-
0.11	command.search.filter	16	-	-
0.02	command.search.fieldalias	16	33,143	33,143
0.02	command.search.index	17	-	-
0.00	command.search.index.usec_1_8	608	-	-
0.00	command.search.index.usec_8_64	37	-	-
0.29	command.search.kv	16	-	-
0.26	command.search.rawdata	16	-	-
0.03	command.search.typer	16	5,021	5,021
0.01	command.search.lookups	16	33,143	33,143
0.00	command.search.track_sourcetypes	17	-	-
0.00	command.search.parse_directives	2	-	-
0.00	command.search.summary	17	-	-
0.00	command.search.tags	16	5,021	5,021
0.53	command.timeliner	18	5,021	5,021
0.00	dispatch.createdSearchResultInfrastructure	1	-	-
0.06	dispatch.evaluate.search	2	-	-
0.31	dispatch.fetch.rcp.phase_0	18	-	-
0.07	dispatch.finalWriteToDisk	1	-	-
0.79	dispatch.localSearch	1	-	-
0.02	dispatch.readEventsInResults	1	-	-
0.78	dispatch.stream.local	17	-	-
0.38	dispatch.timeline	18	-	-
0.00	dispatch.writeStatus	4	-	-
0.11	startup.configuration	2	-	-
0.01	startup.handoff	2	-	-

**Figure 14-10.** Splunk Search Processing Language Query: Execution Cost

- Duration includes the time needed to process each component.
- Component includes the name of the component.
- Invocations include the number of times the component was invoked.
- Input and Output are the events input and output by each component.



Event per second for index="internal" "group=thruput" 33143/1.07=30,94  
 The performance cost query is shown in Figure 14-11.

This search has completed and has returned 5,021 results by scanning 33,143 events in 1.07 seconds  
 (SID: 1575302020.677) [search.log](#)

> Execution costs

Search job properties

canSummarize	None
createTime	2019-12-02T21:23:40.000+05:30
cursorTime	1970-01-01T05:30:00.000+05:30
custom	{ [-] dispatch_earliest_time: -24h@h dispatch_latest_time: now dispatch_sample_ratio: 1 display_page_search_mode: smart search: index="_internal" "group=thruput" workload_pool: }
defaultSaveTTL	604800
defaultTTL	600
delegate	None
diskUsage	1744896
dispatchState	DONE
doneProgress	1
dropCount	None
ea:acl	{ [-] app: search can_write: true modifiable: true owner: admin perms: { [+] } sharing: global ttl: 600 }
earliestTime	2019-12-01T20:30:00.000+05:30
eventAvailableCount	5021
eventCount	5021
eventFieldCount	35
eventIsStreaming	true
eventIsTruncated	None
eventSearch	search (index="_internal" "group=thruput")
eventSorting	desc
indexEarliestTime	1575212430
indexLatestTime	1575302001
isBatchModeSearch	None
isDone	true
isEventsPreviewEnabled	None
isFailed	None
isFinalized	None
isPaused	None
isPreviewEnabled	true
isRealTimeSearch	None
isRemoteTimeline	None
isSaved	None
isSavedSearch	None
isTimeCursored	true

Figure 14-11. Splunk Search Processing Language Query: Performance Cost

- Performance cost displays the status of the search result, provides the event count, and ensures that an events search is performed on all indexes.

In the next section, you learn how the Splunk admin can use `_internal` index to troubleshoot violations due to improper connections between the license master and slave nodes.

## Troubleshooting License Violations

License violation occurs when you exceed the indexing volume allowed for your license. The license master keeps track of license usage. If license usage exceeds license capacity, Splunk issues a license violation warning. If you get five warnings in a rolling 30-day period, you violate the license.

What happens when you violate your license?

During the license violation period, Splunk software continues indexing data but **using search** is blocked, restriction is also imposed on scheduled searches & reports and searching the **\_internal** index is not blocked.

To get detailed license usage report, use `_internal` index and `licenseusage.log`.

## Violation Due to an Improper Connection Between License Master and Slave Node

- A license master defines the license stack and pool, adds licenses, and manages licensing through a central location. The slave nodes report to the master node for licensing distribution.
- A **license slave** is a Splunk Enterprise node that reports to the master for access to Enterprise licenses.

To better understand the link between a license master and license slaves and how to troubleshoot if syncing is absent, you need to refer to the following steps.

- Check the last time the Splunk master contacted a license slave by going to Settings ► Licensing.

- To determine if the master is down, go to the search command in a license slave and type a query similar to the following code block.

```
index="_internal" "license master"
```

- If your master is down, the message's structure looks like the following code block.

```
12/1/19
5:17:10.335 PM12-01-2019 17:17:10.335 +0000 ERROR LMTracker -
failed to send rows, reason='Unable to connect to license master=h
ttps://191.###.####.####:8089 Error connecting: Connect Timeout
```

- If you cannot find a message like the one in the preceding code block, restart your Splunk instance and check for a firewall policy; something in your network is wrong.
- If you find the message in the preceding code block, try to restart the license master or configure a secondary master node. For 72 hours, there is no issue with data to be indexed, but after 72 hours, Splunk deployment is severely affected.

Next, let's look at troubleshooting deployment issues.

## Troubleshooting Deployment Issues

Splunk deployment consists of forwarders that send data to indexers. Indexers index and save the data sent by forwarders. Search heads search data saved in a Splunk indexer. The forwarders, indexers, and search heads form Splunk deployment. Splunk deployment is complex with thousands of forwarders and multiple indexers, or it can have few forwarders, indexers, and search heads—it depends on the scale. It's easy to troubleshoot a simple deployment with only a few forwarders, but to troubleshoot in an environment with thousands of forwarders is very complex. Let's look at troubleshooting forwarders.

To troubleshoot syncing between forwarders and indexers, check the [tcpout] stanza in outputs.conf file in the forwarder.

## Troubleshooting Splunk Forwarders

Splunk forwarders are instances that forward data to a Splunk indexer or heavy forwarder. To troubleshoot a Splunk forwarder, you need to go to the forwarder's `splunkd.log` file at `$(SPLUNK_HOME)/var/log/splunk/splunkd.log`.

The following is the structure statement for a cooked connection.

```
5:17:10.335 PM12-01-2019 17:17:10.335 +0400 warn TcpOutputProc - Cooked
connection to ip=191.x.x.x:9997 timed out
```

- **Timestamp:** The event consists of a timestamp that logs when the event occurred in the system.
- **TcpOutputProc:** A `TcpOutputProc` cooked connection is used when several farms in the query are busy.

How do you troubleshoot `TcpOutputProc`?

- Linux forwarders use `netstat`, `iptables`, and `Telnet` commands to troubleshoot and check whether ports are open; if not, then enable a firewall policy that allows the port for outbound connection.
- Windows forwarders use `Telnet` commands to check whether ports are open; if not, then enable a firewall policy that allows the port for an outbound connection.
- Check the `output.conf` file for syntax. Restart the Splunk forwarder instance after editing the configuration file in the forwarder.

## Troubleshooting Splunk Indexers

Splunk indexers are a repository directory where Splunk saves its processed data. A Splunk indexer receives data from the forwarder if the internal queue on the receiving indexer is blocked in such a scenario. To troubleshoot Splunk indexers, you need to go to `splunkd.log` file of indexer located at `$(SPLUNK_HOME)/var/log/splunk/splunkd.log`.

The following is the structure of statement for `TcpInputFd`.

```
5:19:20.355 PM12-01-2019 17:17:10.335 +0400 ERROR TcpInputFd - SSL Error
for fd from HOST:localhost.localdomain, IP:127.0.0.1, PORT:42134
```

- **Timestamp:** The event consists of a timestamp that logs when the event occurred in the system.
- **TcpInputFd:** A TcpInputFd error is used when several farms in the receiver are busy.

How do you troubleshoot TcpInputFd?

1. Disable receiving on Splunk Web for the port whose farm is busy.
  2. Disable receiving by editing [splunktcp][stanza from][inputs.conf].
- Restart the Splunk instance

Next, let's troubleshoot clustering issues.

## Troubleshooting Clustering Issues

In a Splunk cluster, there are search heads, masters, and indexers. If a cluster is multisite, each cluster has a component. Although the master node resides in any physical site, the master node isn't part of any site.

- The **master node** is responsible for managing the entire cluster. It coordinates the peer nodes' replicating activity and guides the search head in finding data in the peer node.
- The **peer node** receives and indexes incoming data. It works as a standalone machine, but there is one difference: a peer node replicates data coming from another node. The peer node is responsible for storing internal data and copying received data from another node.
- The **search head** runs searches across the set of peer nodes. The master node helps the search head find the data. Search heads are used to search across the entire Splunk environment.

Troubleshooting clustering is complex because there are a series of operations involved

To troubleshoot clustering issues in Splunk, the admin generally uses a Monitoring Console to check for any issues.

## Multi-Search Affinity

When you use multisite clustering, search heads are configured to get search results from a peer node on a local site; this is called multisearch affinity. If a local peer fails and search head goes down at the same time, you won't be able to search for data. To troubleshoot multisearch affinity, disable it by going to `server.conf` in your search head located at `$Splunk_HOME/etc/system/local`. Edit the `server.conf` file as follows.

```
[general]
site = site0

[clustering]
multisite = true
```

## More Bucket Issues

In Splunk, data is grouped into buckets. As the number of buckets increases, time must be increased. If your organization has a bucket that exceeds 38,000 and a peer node greater than 40, it's likely time for you to increase the heartbeat interval time for smoother performance. But before you increase the heartbeat time, consult the Splunk support team to determine whether it's safe for your organization. To increase `heartbeat_timeout`, go to the master node's `server.conf` file located at `$Splunk_HOME/etc/system/local`. Edit the `server.conf` file as follows.

```
[clustering]
heartbeat_timeout=120
```

## Summary

This chapter covered Monitoring Console and all its important components, including single instance and multi-instance deployment, monitoring system health, and forwarder monitoring. It gave an overview of the `metrics.log` file, and you saw how a job inspector troubleshoots a Splunk search. At the end of the chapter, you saw various methods to troubleshoot license violations, deployment issues, and clustering issues in Splunk.

This chapter covered the importance of the Splunk architect exam blueprint. You are now familiar with the 5% of Module 8 (Splunk troubleshooting methods and tools), 5% of Module 12 (search problems), 5% of Module 13: (deployment problems), and 5% of Module 9 (clarifying problems).

The next chapter covers deploying apps through a deployment server, creating a server group using `serverclass.conf`, deploying an app on the universal forwarder through a deployment server, load balancing, the SOCKS proxy, and indexer discovery.

## Multiple-Choice Questions

- A. How does the Monitoring Console monitor forwarders?
  1. by pulling internal logs from forwarders
  2. by using the forwarder monitoring add-on
  3. with internal logs forwarded by forwarders
  4. with internal logs forwarded by the deployment server
- B. How do you remove missing forwarders from the Monitoring Console?
  1. by restarting Splunk
  2. by rescanning active forwarders
  3. by reloading the deployment server
  4. by rebuilding the forwarder asset table
- C. What is the default expiration time of a search job ID?
  1. 10 minutes
  2. 60 minutes
  3. 7 minutes
  4. 100 seconds

- D. Which log files are useful in troubleshooting? (Select all options that apply.)
1. `_internal`
  2. `metrics.log`
  3. `_introspection`
  4. `_license`
- E. The Monitoring Console is useful in troubleshooting clustering issues.
1. `true`
  2. `false`
- F. Which of the following indexes come preconfigured with Splunk Enterprise? (Select all that apply.)
1. `_license`
  2. `_internal`
  3. `_external`
  4. `_fishbucket`

**Answers**

- A. 3
- B. 4
- C. 1
- D. 2
- E. 1
- F. 2, 4



## References

- [www.infoq.com/news/2015/09/Indexer-Cluster/](http://www.infoq.com/news/2015/09/Indexer-Cluster/)
- <https://docs.splunk.com/Documentation/Splunk/8.0.0/SearchReference/Cluster>
- <https://docs.splunk.com/Documentation/Splunk/8.0.0/SearchReference/Cluster>
- [https://wiki.splunk.com/Community:Troubleshooting\\_Monitor\\_Inputs](https://wiki.splunk.com/Community:Troubleshooting_Monitor_Inputs)
- <https://wiki.splunk.com/Deploy:Troubleshooting>
- <https://docs.splunk.com/Documentation/Splunk/8.0.0/SearchReference/Cluster>

## CHAPTER 15

# Advanced Deployment in Splunk

In this chapter, you start a journey of deploying apps through deployment servers. You create app directories and push them to the client through forwarder management, push applications on a client using a server class in forwarder management, and learn how load balancing is implemented in Splunk based on time and volume. You also learn about the SOCKS proxy and indexer discovery.

This chapter covers the following topics.

- Deploying an app through a deployment server
- Creating a server group using `serverclass.conf`
- Deploying an app using a cluster master
- Deploying a search app using a deployment server
- Load balancing
- The SOCKS proxy
- Indexer discovery

By the time you complete this chapter, you will have learned 16% of the Splunk architect exam blueprint.

## Deploying Apps Through the Deployment Server

In this section, you learn how to add deployment apps to the deployment server.

A deployment app in Splunk consists of arbitrary content that you want to download to deployment clients. Arbitrary content can be a Splunk Enterprise app, a set of Splunk configurations, and scripts. You can create a directory for the app. After creating the directory, you can add scripts, configuration files, images, and so forth.

## Create App Directories and View Apps in Forwarder Management

Creating app directories and viewing apps in forwarder management are the most important tasks in Splunk.

### Create App Directories

To create directories in the deployment server for each deployment app, go to the `$SPLUNK_HOME/etc/deployment-apps` folder. There, create a directory for your app. (In this case, I created the `test3` app at `$SPLUNK_HOME/etc/deployment-apps`). The name of the directory serves as the name of the app within forwarder management. You can add apps in Splunk forwarder management at any time. After creating an app, you can run it on Splunk Web or run a CLI command. The following CLI command adds a Splunk app to forwarder management.

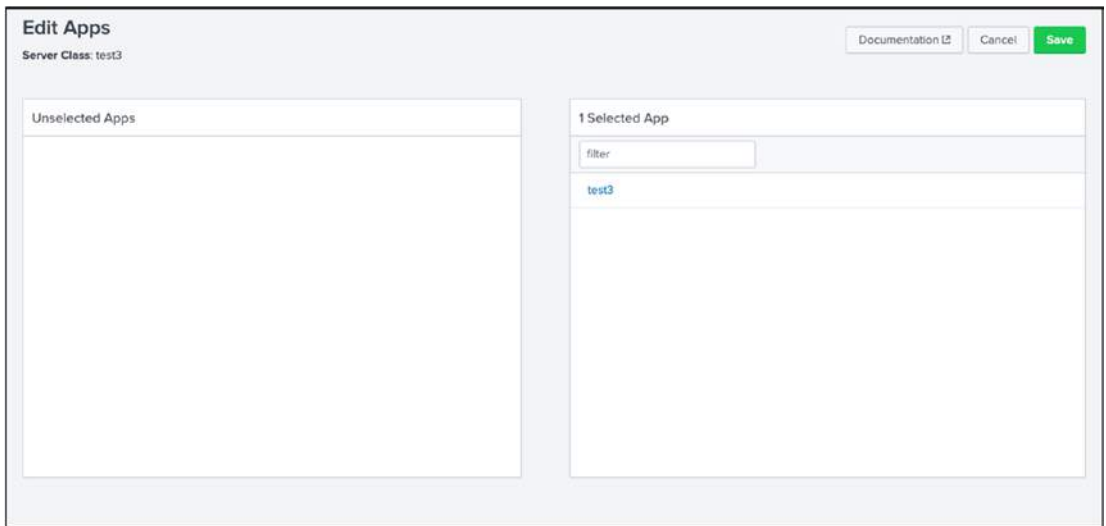
```
splunk reload deploy-server
```

### View Apps in Forwarder Management

Forwarder management plays the most important role in Splunk deployment. Forwarder management provides an interface to create server classes. It maps deployment clients to deployment apps. To view apps in forwarder management, refer to the following steps.

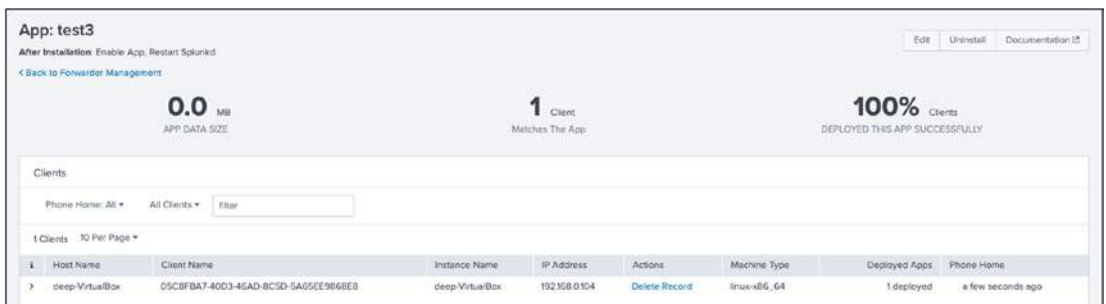
1. Go to the Settings page from Splunk Web and go to Forwarder Management.
2. In Apps, you can find the app name. You should see the app named `test3`. (If not, then in Splunk CLI, type the following command; also refer to [Figure 15-1](#).)

```
splunk reload deploy-server
```



**Figure 15-1.** Forwarder Management

3. Go to app test3. In Actions, click Edit. (See Figure 15-2.)
4. Select the Enable Installation check box, and if you want to restart Splunk, also select the Restart Splunkd check box.
5. Click Save.



**Figure 15-2.** Deploying test3 app using forwarder Management

## Redeploy Apps to the Client

The deployment server is a centralized configuration manager that is responsible for deploying apps on client instances. Redeployment is generally done when you change the server class (changes in the set of apps or clients), new clients join a server class, or change an app's configuration file. Generally, redeployment is done automatically, but in some cases, you need to redeploy using `serverclass.conf` or an app using forwarder management.

### Deploying an App Using Forwarder Management

The server class determines the content that needs to be deployed to a group of clients. When you create a server class in Splunk, you tell the deployment server that a group of clients should receive configurations in the form of apps.

### Deploy an App by Editing `serverclass.conf`

You can also create server classes by editing the `serverclass.conf` file. The deployment server does not automatically deploy the app to the client. To deploy an app to the client, you need to manually configure forwarder management by writing the following CLI command.

```
splunk reload deploy-server
```

### Redeploy an App After You Change the Content

When you add a script or a configuration file to your existing app in Splunk, it's necessary to reflect the changes in all your client machines. To redeploy an app with updated content, you need to follow these instructions.

1. Update the content in the deployment app directory on the deployment server.
2. Reload the deployment server.

### Deploy Apps to New Client

When a new client is added to the deployment server list, the deployment server automatically deploys all apps in the client machine, which passes the deployment server's filter. It is done automatically if you use forwarder management; otherwise, you need to manually configure forwarder management by writing a CLI command.

## App Management Issues

The deployment server is very useful for managing apps, but there are some issues linked to app management. Before you deploy an app, let's discuss issues linked to app management in Splunk.

### Deployment Server Is Irreversible

Once you start using a deployment server to manage apps, you cannot stop using it as when an app is removed from the deployment server, the client behavior is to delete its instances but deleting the app from the deployment server is not telling a client to stop using the deployment server to manage that app; it's just telling it to uninstall the application and there is no way to tell the deployment client to manage the app on its own.

### Apps with Lookup Tables

Indexers or search heads might contain running apps that store information in lookup tables. The deployment server manages apps and distributes updates in the apps. If there is an update in an app configuration, then it overwrites any existing updates, at which time there is a chance that you lose your lookups.

## Creating a Server Group Using `ServerClass.conf`

Forwarder management in Splunk provides an interface to create server classes and map deployment clients to deployment apps. The interface saves server configurations to the `serverclass.conf`. In some advanced server class configurations, you might need to directly edit `serverclass.conf`.

## Configure a Server Class Using Forwarder Management

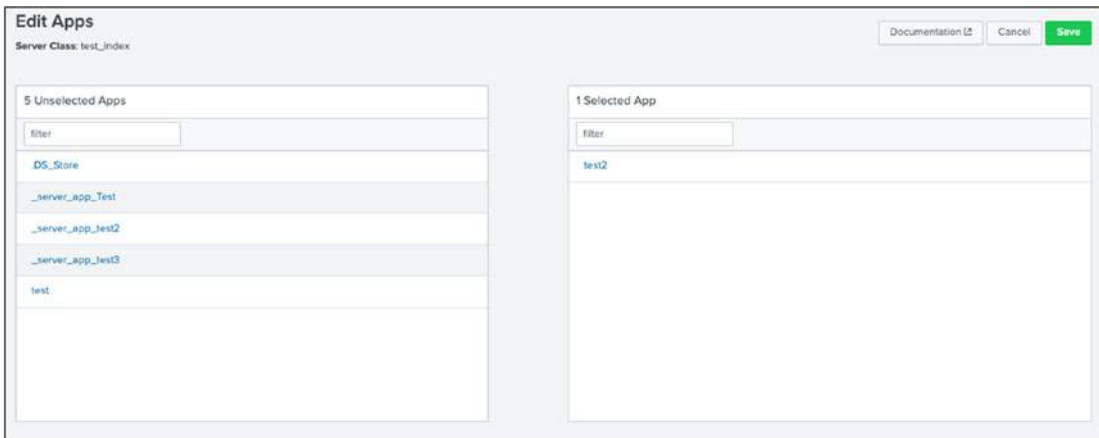
Server classes map deployment clients to deployment apps. To create a server class, refer to the following steps.

1. Go to Settings and go to Forwarder Management.
2. Go to server Classes and go to New Server Class.
3. In Server Class, set Name=test\_index (see Figure 15-3).



**Figure 15-3.** Server Class:test\_index

4. Go to Add Apps and select the apps you want to deploy on deployment clients (see Figure 15-4).



**Figure 15-4.** test\_index:deployment apps

5. Go to Add Clients, and in the client, you find a whitelist and blacklist block, followed by all clients that are polling to the deployment server.
  - **Whitelist in adding apps:** This is a required field in which you must provide the hostname, IP, or DNS.
  - **Blacklist in adding apps:** This is an optional field where you can provide the hostname, IP, or DNS of what you want to exclude. (See Figure 15-5.)

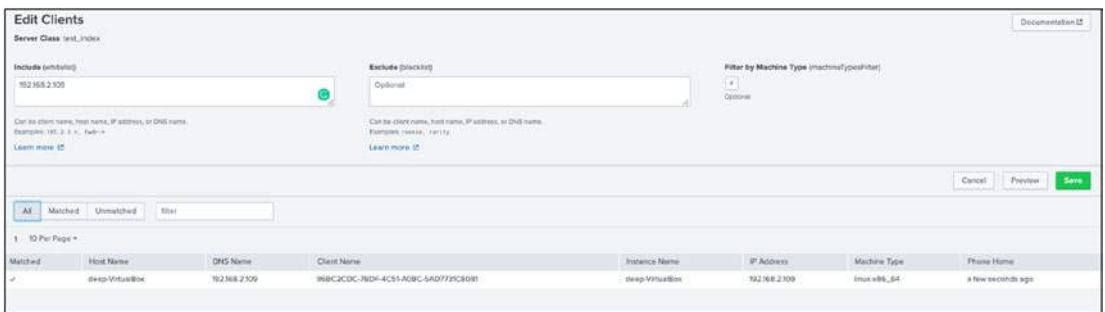


Figure 15-5. *test\_index:deployment clients*

## Deploy Configuration File Through Cluster Master

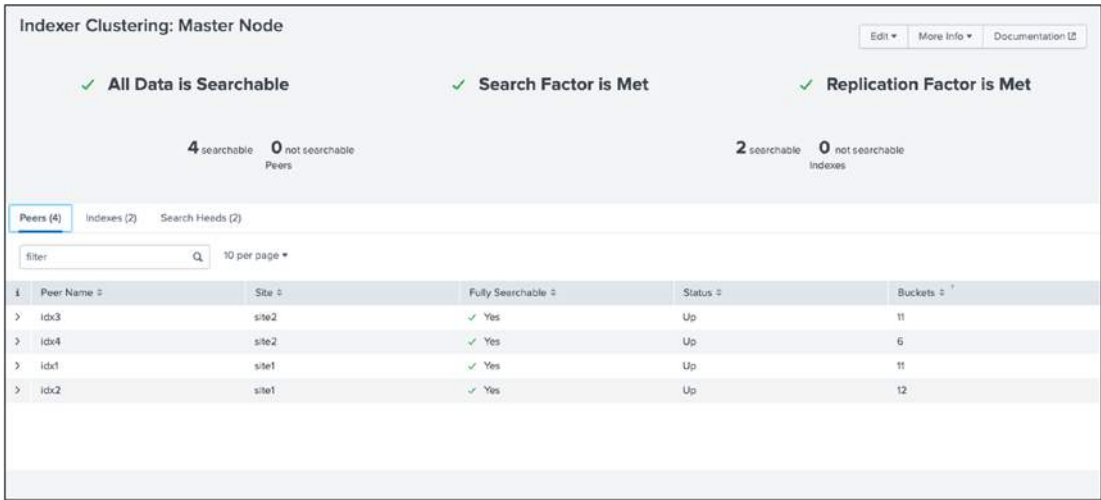
The cluster master node in Splunk coordinates the peer node activities that regulate the functioning of the indexer cluster. The master node controls and manages index replication, distributes app bundles to peer nodes, and tells the peer nodes which peer to search. (You learned how to configure the master node in Chapter 9.)

## Managing Indexes on Indexer Using Master Node

To manage the indexes of all peer nodes, you need to use the same configuration file for all indexers. You do not need to go to each of the Splunk instances and edit its indexes file (it is not recommended to manually edit each indexer's configuration file). You need to use the master node for the deployment of the same configuration file for all indexers. In this section, you deploy the same indexes.conf file on four indexers with two sites—site1 and site2—using the cluster master.



On the master node (using Splunk Web), go to settings and move to Indexer Clustering. There, you can see all the indexes that have been reported to the master node. In my case, four indexers reported to the master node. Figure 15-6 shows the instances that pinged the master node.



**Figure 15-6.** Master Node:Indexer Clustering

An index named *test* was configured in Chapter 1. This index needs to be configured on all three indexers. You could configure it manually, but it is not a good practice. In this section, you use a cluster master to deploy the *index.conf* file, where you enable a replication factor and configure the index.

1. Go to the cluster master node. Traverse to `$SPLUNK_HOME/etc/cmaste/_local` and create *indexes.conf* file.
2. Edit *indexes.conf* according to the following code.

```
[test]
homePath=$SPLUNK_DB/test/db
coldPath=$SPLUNK_DB/test/colddb
thawedPath=$SPLUNK_DB/test/thaweddb
repFactor=auto
```

- To deploy indexes.conf file, migrate to \$SPLUNK\_HOME/bin/ in the cluster master.

```
sudo ./splunk validate cluster-bundle --check restart
```

- To confirm the status of bundle validation, refer to the following command.

```
sudo ./splunk show cluster-bundle-status
```

If your cluster bundle is deployed successfully, you find the checksum of your deployment. The command is similar to Figure 15-7.

```
[ec2-user@ip-172-31-75-109 bin]$ sudo ./splunk validate cluster-bundle --check restart
Validating new bundle. Please run 'splunk show cluster-bundle-status' to check the status of the bundle validation.
[ec2-user@ip-172-31-75-109 bin]$ sudo ./splunk show cluster-bundle-status

master
  cluster_status=None
  active_bundle
    checksum=826AF3010CA7165419661216C27A2AAD
    timestamp=1592876799 (in localtime=Tue Jun 23 01:46:39 2020)
  latest_bundle
    checksum=826AF3010CA7165419661216C27A2AAD
    timestamp=1592876799 (in localtime=Tue Jun 23 01:46:39 2020)
  last_validated_bundle
    checksum=826AF3010CA7165419661216C27A2AAD
    last_validation_succeeded=1
    timestamp=1592918946 (in localtime=Tue Jun 23 13:29:06 2020)
  last_check_restart_bundle
    last_check_restart_result=restart not required
    checksum=
    timestamp=0 (in localtime=Thu Jan 1 00:00:00 1970)

idx3  0700F4C5-9908-4036-AE0C-58A347989BAA  site2
  active_bundle=826AF3010CA7165419661216C27A2AAD
  latest_bundle=826AF3010CA7165419661216C27A2AAD
  last_validated_bundle=826AF3010CA7165419661216C27A2AAD
  last_bundle_validation_status=success
  restart_required_apply_bundle=0
  status=Up

idx4  DC29F3C9-32C4-454C-8D53-8CDDAA618549  site2
  active_bundle=826AF3010CA7165419661216C27A2AAD
  latest_bundle=826AF3010CA7165419661216C27A2AAD
  last_validated_bundle=826AF3010CA7165419661216C27A2AAD
  last_bundle_validation_status=success
  restart_required_apply_bundle=0
  status=Up

idx1  DCD95AC2-E49C-4CEB-A9D4-C54B00C2B832  site1
  active_bundle=826AF3010CA7165419661216C27A2AAD
  latest_bundle=826AF3010CA7165419661216C27A2AAD
  last_validated_bundle=826AF3010CA7165419661216C27A2AAD
  last_bundle_validation_status=success
  restart_required_apply_bundle=0
  status=Up

idx2  FEB0AD2C-4FEF-43DC-83D1-3230BFC080EE  site1
  active_bundle=826AF3010CA7165419661216C27A2AAD
  latest_bundle=826AF3010CA7165419661216C27A2AAD
  last_validated_bundle=826AF3010CA7165419661216C27A2AAD
  last_bundle_validation_status=success
  restart_required_apply_bundle=0
  status=Up
```

Figure 15-7. Deploying cluster bundle on peer indexer

Confirm whether `repFactor=auto` is set up for the test index. To monitor the cluster master status, navigate to Splunk Web settings and go to Indexer Clustering, where you find the index test’s status (see Figure 15-8).

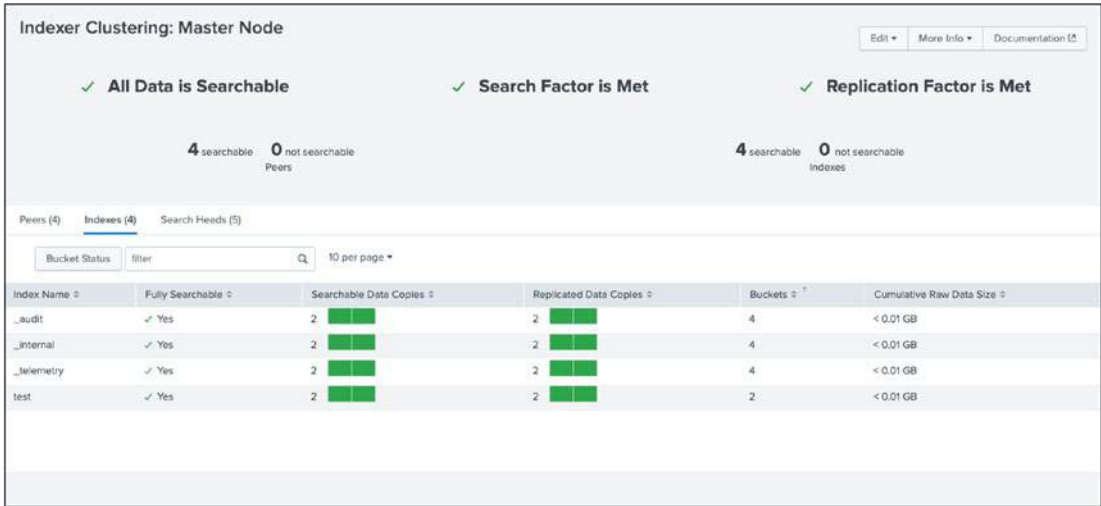


Figure 15-8. Indexer Clustering:test index

## Deploy App on Search Head Clustering

The deployer distributes apps to the search head cluster. The deployer is a Splunk instance that distributes apps and configuration updates to each search head cluster member. The set of updates that the deployer distributes is called **configuration bundles**. The main task of the deployer is to handle the migration of apps and user configuration to search head clusters from non-cluster instances, search head pools, deploy baseline app configuration to search head cluster members, and provide means to distribute non-replicated, non-runtime configuration updates to all search head cluster members.

### Configure the Deployer to Distribute Search Head Apps

In this section, you use a deployer to push the shc (search head cluster) app to search head clustering nodes. You already saw how to configure search head clustering. In this section, you learn how to deploy the solution.

1. To deploy shc app using a deployment server, migrate to `$SPLUNK_HOME/etc/local/server.conf` in Search Heads. Refer to the following command.

```
[shclustering]
pass4SymmKey = <key>
```

2. Restart the Splunk instance after making changes.

```
splunk restart
```

3. To deploy shc app using a deployment server, migrate to `$SPLUNK_HOME/bin/` of search heads. Refer to the following command.

```
sudo ./splunk edit shcluster-config -conf_deploy_fetch_url
<url>:<mgmt_port>,<host_name>:<mgmt_port>
```

4. In the deployer, create an app named shc located in `$SPLUNK_HOME/etc/shcluster/apps/`
5. Create or edit `apps.conf` located in `/local/apps.conf` of the shc app. Refer to the following code.

```
[ui]
is_visible = 0
[package]
id = shc_base
check_for_updates = 0
```

6. Create or edit `outputs.conf` located in `/local/outputs.conf` of the shc app. Refer to the following code.

```
[indexAndForward]
index = false

[tcput]
defaultGroup = default-autolb-group
forwardedindex.filter.disable = true
indexAndForward = false
```

```
[tcpout:default-autolb-group]
server=<url1>:<mgmt_port>/<host_name1>:<mgmt_port>,<url2>:<mgmt_
port>/<host_name2>:<mgmt_port>,.....,<url3>:<mgmt_
port>/<host_name3>:<mgmt_port>
```

- 7. To push the configuration in search head clustering nodes, migrate to \$SPLUNK\_HOME/bin/ in the deployment server and refer to the following command. (see Figure 15-9).

```
splunk apply shcluster-bundle -action stage --answer-yes
```

```
[[root@ip-172-31-77-218 bin]# sudo ./splunk apply shcluster-bundle -action stage --answer-yes
Your session is invalid. Please login.
[Splunk username: admin
[Password:
[Bundle has been pushed successfully to all the cluster members.
[[root@ip-172-31-77-218 bin]# █
```

**Figure 15-9.** Deploying cluster bundle on Search Head Clustering

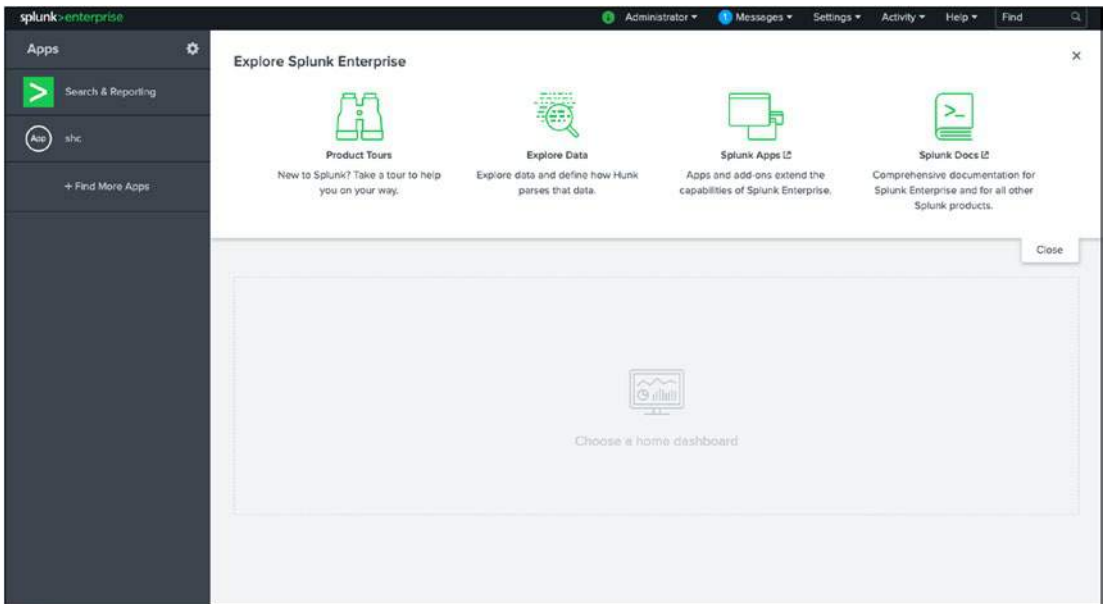
- 8. The deployment is ready to be pushed. To do this, you need to contact the search head cluster captain and use the apply shcluster-bundle command. Refer to the following command. (see Figure 15-10).

```
splunk apply shcluster-bundle -action send -target
<url>:<mgmt_port>/<host_name>:<mgmt_port> --answer-yes
```

```
[[root@ip-172-31-77-218 bin]# sudo ./splunk apply shcluster-bundle -action stage --answer-yes
Your session is invalid. Please login.
[Splunk username: admin
[Password:
[Bundle has been pushed successfully to all the cluster members.
[[root@ip-172-31-77-218 bin]# sudo ./splunk apply shcluster-bundle -action send -target https://172.31.72.220:8089 --answer-yes
[Bundle has been pushed successfully to all the cluster members.
[[root@ip-172-31-77-218 bin]# █
```

**Figure 15-10.** Deploying cluster bundle on Search Head using Search Head Cluster Captain

- 9. When you traverse to the search head clustering nodes, you can find shc app deployed on it. Refer to Figure 15-11. Similarly, you can find the app deployed.



*Figure 15-11. Search Head:SHC App*

## Load Balancing

A forwarder forwards data to the Splunk indexer. A forwarder distributes data across several receiving instances. Every receiver gets a portion of the total data. The forwarder routes data to the total available indexers at a specified time or volume interval. The choices vary according to your needs. If you have three available indexers—indexer1, indexer2, and indexer3—and a group of forwarders, the forwarder switches the data stream to another index in a site at random based on time or volume.

- **By time:** This factor controls how frequently forwarders switch from one indexer to another. To enable the load balancer by time, update the settings in the universal forwarder's `outputs.conf` file by adding `autoLBFrequency=<number>`.
- **By volume:** This factor controls how frequent forwarders switch from one indexer to another. To enable the load balancer by volume, update the settings in the universal forwarder's `outputs.conf` file by adding `autoLBVolume=<number>`.

## Configure Static Load Balancing in Splunk Forwarder Using outputs.conf

To configure static load balancing in a Splunk forwarder using outputs.conf, you need to go to \$SPLUNK\_HOME/etc/system/local and edit outputs.conf. (If the file doesn't exist, you need to create it.) You can configure the load balancer either by time or by volume.

### Configure a Static Load Balancer by Time

Go to \$SPLUNK\_HOME/etc/system/local/outputs.conf on the forwarder and edit the stanza according to the following configuration, which sets autoLBFrequency by time. This factor controls how frequently forwarders switch from one indexer to another indexer. Refer to the syntax for configuring a static load balancer by time using the following configuration file.

```
[tcpout:my_LB_indexers]
server=<ip1>:<port>/<host_name1>:<port>,<ip2>:<port>/<host_name2>:<port>,,,
,,,,<ipn>:<port>/host_namen>:<port>
autoLBFrequency=<number>
```

- server is the address of indexer.
- autoLBFrequency is a time factor in seconds that allows forwarders to switch from one indexer to another indexer.

### Configure Static Load Balancer by Volume

Go to \$SPLUNK\_HOME/etc/system/local/outputs.conf on the forwarder and edit the stanza according to the following configuration, which sets autoLBVolume by volume. This factor controls how frequently forwarders switch from one indexer to another. Refer to the syntax for configuring static load balancer by time using the following configuration file.

```
[tcpout:my_LB_indexers]
server=<ip1>:<port>/<host_name1>:<port>,<ip2>:<port>/<host_name2>:<port>,,,
,,,,<ipn>:<port>/host_namen>:<port>
autoLBVolume=<number>
```

- **server** is the address of indexer.
- **autoLBVolume** is a volume factor in bytes that allows forwarders to switch from one indexer to another.

## Specify Load Balancing from Splunk CLI

To configure load balancing from Splunk CLI, you need to go to \$SPLUNK\_HOME/bin on the forwarders and specify the commands. It is the easiest way to load balance.

To specify load balancing from Splunk CLI, refer to the following.

```
splunk add forward-server <ip>:<port>/<host>:<port> -method autobalance
splunk restart
```

## Indexer Discovery

Indexer discovery streamlines the process of connecting forwarders to peer nodes within indexer clusters. Indexer discovery comprises peer nodes, master nodes, and forwarders. The peer node in indexer discovery provides the master node with information about their receiving ports. The forwarder “polls” master nodes at regular intervals for a list of the peer nodes. The master node (in response to forwarder requests) provides the address of the available peer nodes according to load balancing policies. The forwarder starts sending data to the peer node.

## Configure Indexer Delivery

To configure indexer delivery successfully, you need to configure the components. In indexer discovery, peer nodes, master nodes, and forwarders are the main components. You need to configure each component as follows.

1. Configure the peer nodes to receive data from forwarders.
2. Configure the master node to enable indexer discovery.
3. Configure the forwarders.



## Configure the Peer Nodes

You can configure it either by editing the configuration files or by using Splunk Web. Refer to the following instructions.

### Configure Peer Node Using Splunk Web

To configure peer nodes to receive data from forwarders using Splunk Web, follow these steps.

1. Go to Splunk Web, go to Settings, and move to Forwarding and Receiving.
2. In Configure Receiving, click Add New.
3. In Port, type **port address<port>**.
4. Click Save.

### Configure the Peer Node Using `inputs.conf`

To configure peer nodes to receive data from forwarders using a configuration file, go to `$SPLUNK_HOME/etc/system/local/inputs.conf` and edit `splunktcp` stanza. Refer to the following `splunktcp` stanza.

```
[splunktcp://<port>]
disabled = 0
```

### Configure the Master Node

To configure the master node to enable master node discovery using a configuration file, go to `$SPLUNK_HOME/etc/system/local/server.conf` and edit the `indexer_discovery` stanza. Refer to the following.

```
[indexer_discovery]
pass4SymmKey = <string>
polling_rate = <integer>
indexerWeightByDiskCapacity = <bool>
```

- **pass4SymmKey** in the master node secures the connection between forwarders and the master node.
- **polling\_rate** adjusts the rate at which the forwarders poll the master node.
- **indexerWeightByDiskCapacity** is an optional attribute that determines whether indexer discovery uses load balancing.

## Configure the Forwarders

To configure the forwarders using a configuration file, go to `$SPLUNK_HOME/etc/system/local/outputs.conf` and edit the `indexer_discovery:<name>` stanza. Refer to the following.

```
[indexer_discovery:<name>]
pass4SymmKey = <string>
master_uri = <uri>

[tcput:<target_group>]
indexerDiscovery = <name>

[tcput]
defaultGroup = <target_group>
```

- **indexer\_discovery:<name>** in the forwarder stanza sets `<name>` in the indexer discovery attribute.
- **pass4SymmKey** secures the connection between forwarders and master nodes.
- **master\_uri** contains the address of the master node.
- **tcput:<target\_group>** sets the indexer discovery attribute instead of the server attribute you use to specify the receiving peer nodes if you are not enabling indexer discovery.

## SOCKS Proxy

The forwarder directly communicates with the indexer and sends data, but if the firewall blocks connectivity, the forwarder cannot communicate with the indexer. In this scenario, you can configure a forwarder to use a SOCKS5 proxy host to send data to an indexer by specifying attributes in a stanza in the forwarder's `outputs.conf` file. The proxy host establishes a connection between the indexer and forwarder; then, the forwarder sends data to the indexer through the proxy host.

### Configure SOCKS Proxy

A SOCKS proxy is configured by creating a `[tcpout]` or `[tcpout-server]` stanza within the `output.conf` file in the forwarder's `$SPLUNK_HOME/etc/system/local/outputs.conf` file. Restart the forwarder after the configuration is updated.

Table 15-1 illustrates the configuration of a SOCKS proxy in the forwarder's `outputs.conf` file.

**Table 15-1.** *Configure SOCKS Proxy using forwarder's outputs.conf*

Attribute	Value
<code>socksServer=&lt;ip&gt;:&lt;port&gt;, &lt;host_name&gt;:&lt;port&gt;</code>	Specifies the address of socks5 proxy where forwarder should connect for forwarding data.
<code>socksUsername=&lt;string&gt;</code>	Specifies the username to authenticate socks5 proxy. It is optional.
<code>socksPassword=&lt;string&gt;</code>	Specifies the password to authenticate socks5 proxy. It is optional.
<code>socksResolveDNS=truelfalse</code>	Specifies whether forwarder should use DNS to resolve the hostname of indexer in the output group.

The following code block illustrates an example of how a SOCKS proxy is configured in Splunk.

```
[tcpout]
defaultGroup = proxy_indexer_test

[tcpout:proxy_indexer_test]
server = <hostname1>:<port>,<hostname2>:<port>,...,<hostname n>:<port n>
socksServer = <hostname>:<port>
```

- **server** is the name of the server.
- **socksServer** specifies the address of socks5 proxy where the forwarder should connect to forward data.

## Summary

In this chapter, you started the journey of deploying an app through a deployment server. You created app directories, deployed an app to the client through the deployment server, and pushed it to the client through forwarder management. You created a server group using `serverclass.conf` and pushed applications on a client using the server class in forwarder management. You also deployed an app on universal forwarder through the deployment server. In the next section, you learned how load balancing is implemented on Splunk based on time and volume. In a SOCKS proxy, you learned that if a firewall is not allowed to pass data through the network, you can implement a SOCKS proxy. You also learned how indexer discovery is implemented to streamline the process of connecting forwarders to peer nodes in indexer clusters.

In this chapter, you covered a hefty portion of the Splunk architect exam blueprint. You are familiar with the 6% of Module 6 (forwarders and deployment best practices), 5% of Module 19 (search head cluster management and administration), and 5% of Module 14 (large scale splunk deployment).

In the next chapter, you learn about managing indexes, managing index storage, managing index clusters, managing a multisite index cluster, Splunk REST API endpoints, and Splunk SDK.

## Multiple-Choice Questions

- A. Where should apps be located on the deployment server that the clients pull from?
1. \$SPLUNK\_HOME/etc/apps
  2. \$SPLUNK\_HOME/etc/search
  3. \$SPLUNK\_HOME/etc/master-apps
  4. \$SPLUNK\_HOME/etc/deployment-apps
- B. Which Splunk component distributes apps and certain other configuration updates to search head cluster members?
1. deployer
  2. cluster master
  3. deployment server
  4. search head cluster master
- C. When running the following command, what is the default path in which deployment server.conf is created?
- ```
splunk set deploy-poll deploy server:port
```
1. SPLUNK\_HOME/etc/deployment
  2. SPLUNK\_HOME/etc/system/local
  3. SPLUNK\_HOME/etc/system/default
  4. SPLUNK\_HOME/etc/apps/deployment
- D. Which methods are used in load balancing? (Select all methods that apply.)
1. by time
  2. by frequency
  3. by volume
  4. all of the above

- E. What are the main components of index discovery in Splunk?  
(Select all methods that apply.)
1. peer node
  2. master node
  3. universal forwarder
  4. search head
  5. all of the above
  6. none of the above

**Answers**

- A. 4  
B. 3  
C. 2  
D. 1, 3  
E. 1, 2, 3

## References

- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Updating/Updateconfigurations>
- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Updating/Updateconfigurations>
- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Updating/Excludecontent>
- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Updating/Defineddeploymentclasses>
- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Updating/Defineddeploymentclasses>
- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Updating/Defineddeploymentclasses>

- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Updating/Updateconfigurations>
- <https://docs.splunk.com/Documentation/Forwarder/8.0.2/Forwarder/Configureloadbalancing>
- <https://docs.splunk.com/Documentation/Forwarder/8.0.1/Forwarder/ConfigureaforwardertouseaSOCKSproxy>

## CHAPTER 16

# Advanced Splunk

In this chapter, you learn about managing indexes, configuring custom indexes, removing indexes, and configuring index parallelization. You see how to move the index database, configure maximum index size, and set limits for disk usage. You learn how to configure nodes in an offline state, enable maintenance mode, perform a rolling restart, remove bucket copy, and remove peer nodes. You learn what to do if the master site fails. You go through REST API endpoint capabilities in Splunk and look at the Splunk SDK.

This chapter covers the following topics.

- Managing indexes
- Managing index storage
- Managing index clusters
- Managing a multisite index cluster
- Splunk REST API endpoints
- Splunk SDK

By the time you complete this chapter, you will have covered roughly 18% of the Splunk architect exam blueprint.

## Managing Indexes

An index in Splunk is a collection of directories and files. Index directories in Splunk are also known as buckets. They are grouped according to their age; for example, hot buckets, cold buckets, and warm buckets. Index directories are discovered at `$SPLUNK_HOME/var/lib/splunk`. Splunk Enterprise provides support to two types of indexes.



- **Event indexes** impose minimal structure and can occupy any type of data, including metrics data.
- **Metrics indexes** are used in a highly structured format to handle the higher volume and lower latency demands associated with metrics data.

Let's see how to configure event indexes and metrics indexes.

## Configure Event Indexes

You can configure event indexes using Splunk Web, CLI, or a configuration file. If you want to configure your indexes for the entire index cluster, you need to edit it with a configuration file (i.e., go to `SPLUNK_HOME/etc/system/local/indexes.conf`) and edit it directly.

### Configure Event Indexes Using a Splunk Configuration File

To add a new index for event indexes in the Splunk configuration file, go to `$$SPLUNK_HOME/etc/system/local` and edit `indexes.conf`. In the stanza name, provide the name of an indexer. You can refer to the following stanza block for more information on regarding event indexes using a Splunk configuration file.

```
[<index_name>]
homePath=<path for hot and warm buckets>
coldPath=<path for cold buckets>
thawedPath=<path for thawed buckets>
```

- **homePath** defines `<homepath >` in configuring event indexes. Normally, data in warm and hot buckets reside in `<homepath>` in an index.
- **coldPath** defines `<coldpath>` in configuring event indexes. Normally, data in cold buckets reside in `<coldPath>` in an index.
- **thawedPath** defines `<thawedpath>` in configuring event indexes. Normally, data in frozen buckets or archived data reside in `<thawedPath>` in an index.

## Configure Event Indexes Using Splunk CLI

To configure event indexes using Splunk CLI, go to `$SPLUNK_HOME/bin` and refer to the following command to configure event indexes.

```
splunk add index <index_name> -homePath <homePath> -coldPath <coldPath>
    -thawedPath <thawedPath>
```

## Configure Metrics Indexes

To configure metrics indexes in Splunk, you can use Splunk Web, CLI, or edit configuration files. If you want to configure your indexes for the entire index cluster, you need to edit the configuration files. Go to `SPLUNK_HOME/etc/system/local/indexes.conf` and edit it directly. You cannot configure your indexes for the entire index cluster using CLI and Splunk Web.

## Configure Metric Indexes Using a Splunk Configuration File

```
[test]
homePath=<path for hot and warm buckets>
coldPath=<path for cold buckets>
thawedPath=<path for thawed buckets>
datatype=metric
```

- **homePath** defines `<homepath >` in configuring metric indexes. Normally, data in warm and hot buckets reside in `<homepath>` in an index.
- **coldPath** defines `<coldpath>` in configuring metric indexes. Normally, data in cold buckets reside in `<coldPath>` in an index.
- **thawedPath** defines `<thawedpath>` in configuring metric indexes. Normally, data in frozen buckets or archived data reside in `<thawedPath>` in an index.
- **datatype** defines the of index, whether it is metric or event.

To configure event indexes using Splunk CLI, go to `$SPLUNK_HOME/bin` and refer to the following command.

```
splunk add index <index_name> -datatype metric -homePath <homePath>
-coldPath <coldPath> -thawedPath <thawedPath>
```

## Remove Indexes and Index Data for Managing Indexes

Removing indexes and index data includes removing data from the index, removing all the data from Splunk, removing the entire index, and even disabling it. This is possible using Splunk CLI commands.

Refer to the following command to remove all data from all indexes.

```
splunk clean eventdata -index <index_name>
```

Refer to the following command to remove all data from all indexes.

```
splunk clean eventdata
```

Refer to the following command to remove a particular index.

```
splunk remove index <index_name>
```

Disabling an index is a better option than deleting an index. When you disable an index, you can re-enable it if you want. Refer to the following command to disable an index.

```
splunk disable index <index_name>
```

To enable an index in Splunk, refer to the following command.

```
splunk enable index <index_name>
```

## Configure Index Parallelization for Managing Indexes

Index parallelization is a feature that allows an index to maintain multiple pipeline sets. A *pipeline set* handles data processing from the ingestion of raw data, through event processing, and to writing the events to disk. An index runs a single pipeline set. However, if the underlying machine is underutilized in terms of available cores and I/O, you can configure the index to run two pipeline sets. By running two pipeline sets, you potentially double the index's indexing throughput capacity.

## Configure Pipeline Sets for Index Parallelization

To configure pipeline sets for index parallelization, edit `server.conf` if it exists, or else create a `server.conf` file at `$SPLUNK_HOME/etc/system/local` with the following stanza.

```
[general]
parallelIngestionPipelines = 2
```

## Configure the Index Allocation Method for Index Parallelization

To configure the index allocation method in Splunk for index parallelization, you can modify the `server.conf` file located at `$SPLUNK_HOME/etc/system/local`. There are two methods to configure allocation methods in Splunk. It can be either a round-robin selection method or a weighted-random selection method. Refer to the following stanza.

```
[general]
pipelineSetSelectionPolicy = round_robin | weighted_random
```

In the next section, you learn how to manage storage so that indexes operate smoothly.

## Manage Index Storage

Managing index storage is crucial in Splunk. You can configure index storage, move index databases, configure index size, set limits in index storage, and set retirement and archiving policies.

## Move the Index Database

You can move the index database in Splunk from one location to another by changing the path definition of `SPLUNK_DB`.

Follow these steps to move the index database in Splunk.

1. Create the directory with write permissions.
2. Stop the indexer using Splunk CLI.

```
splunk stop
```

3. Copy index file system to the directory.
4. Unset the Splunk environment variable using Splunk CLI.

```
unset SPLUNK_DB
```

5. Change the SPLUNK\_DB attribute in \$SPLUNK\_HOME/etc/splunk-launch.conf to the path you want.

```
SPLUNK_DB=<path>
```

6. Start the indexer using Splunk CLI.

```
splunk start
```

## Configure Maximum Index Size for Indexer Storage

To configure the maximum index size for indexer storage in Splunk, you can use the configuration file. You learn how to set the maximum size for each bucket and set the maximum index size for volume.

To configure the maximum index size for a bucket in Splunk, modify `indexes.conf` in `$SPLUNK_HOME/etc/system/local` and add `maxTotalDataSizeMB` in the configuration file. Refer to the following code block for more help.

```
[<index_name>]
homePath.maxDataSizeMB = <value>
coldPath.maxDataSizeMB = <value>
```

- `<index_name>` is the name of the bucket.
- `homePath.maxDataSizeMB` is the maximum value of the bucket homepath, and the value is in megabytes (MB).
- `coldPath.maxDataSizeMB` stands for the maximum value of the bucket coldpath, and the value is in megabytes (MB).

To configure the maximum index size for volume in Splunk, modify indexes.conf in \$SPLUNK\_HOME/etc/system/local and add maxVolumeDataSizeMB in the configuration file. Refer to the following code block for more help.

```
[volume:<volume_name>]
path = <path>
maxVolumeDataSizeMB = <value>
```

- volume\_name is the name of the volume that you want to configure.
- maxVolumeDataSizeMB is the value of the path that you want to configure.

In configuring volume for each index in Splunk, refer to the following stanza.

```


```

- In configuring volume for buckets, volume\_name stands for the name of the volume
- In configuring a name for buckets, idx\_name stands for the name of the indexer.

## Set Limit for Disk Usage in Splunk

To limit disk usage in Splunk, you can set the minimum amount of free space for the disk where the indexed data is stored. If the limit is reached, the indexer stops indexing data. Generally, the indexer checks the partition that contains indexes. If the limit is reached, the indexer stops indexing.

## Configure Splunk to Set a Limit for Disk Usage Using Splunk CLI

The limit for disk usage is used mainly for disk space issues so that you don't run out of disk space. To configure Splunk to limit disk usage, use Splunk CLI. Go to \$SPLUNK\_HOME/bin and refer to the following commands.

```
splunk set minfreemb <size>
```

To enable configuration, you need to restart the Splunk instance. Refer to the following command.

```
splunk restart
```

## Configure Splunk to Set a Limit for Disk Usage Using a Splunk Configuration File

To configure Splunk to set a limit for disk usage using a Splunk configuration file, go to `$SPLUNK_HOME/etc/system/local` and edit `server.conf`. Refer to the following code block.

```
[diskUsage]
minFreeSpace = <num>
```

**minFreeSpace** sets the limit for disk usage in Splunk. It is followed by a value that indicates how much free space the Splunk index should have.

Now let's move to the next section to learn best practices for index clusters.

## Managing Index Cluster

Managing index clustering in Splunk comprises tasks like taking a peer node offline, using maintenance mode, a rolling restart of index cluster, and removing excess bucket clusters from the index cluster. Let's look at each topic, starting with configuring a peer node to offline.

### Configuring Peer Node to Offline

When you want to take a peer down temporarily or permanently from your existing Splunk environment, the `splunk offline` command has two advantages: you can take out peer nodes temporarily and permanently. When you use the `splunk offline` command, it minimizes search disruption in Splunk.

### Configure Splunk to Offline Mode Using Splunk CLI

Splunk `offline` command handles both types of peer shutdown (i.e., temporary, or permanent).

- **splunk offline** takes a peer offline for maintenance purposes.
- **splunk offline --enforce-counts** permanently takes down a peer from the cluster.

Taking down a peer temporarily is the fastest method after the peer is shut down. The cluster initiates bucket-fixing activities. Refer to the following code block.

```
splunk offline
```

Take down a peer down temporarily after the cluster attempts to reallocate primary copies on the peer. Complete any searches that the peer is currently doing. The value of `decommission_node_force_timeout` determines the maximum time allotted for the primary allocation of buckets in the peer node's `server.conf` file. By default, it's 5 minutes. The peer also waits for any ongoing searches to complete by the value of `decommission_search_jobs_wait_secs` in the peer node's `server.conf` file. By default, it's 3 minutes. To change the time period allocated for the cluster to return to the valid state, refer to the following command.

```
splunk offline --decommission_node_force_timeout <seconds>
```

To change the time allocated for the peer node on the master node's restart period, refer to the following command.

```
splunk edit cluster-config -restart_timeout <seconds>
```

To take down a peer permanently, two things need to be taken care of before the peer node is shut down permanently—bucket fixing activities and the ongoing completion process. Refer to the following command to take down the peer node permanently in Splunk.

```
splunk offline --enforce-counts
```

## Configure Splunk to Maintenance Mode Using Splunk CLI

Maintenance mode halts the most critical bucket fixup activities and prevents the rolling of hot buckets. Maintenance mode is useful when performing peer and other maintenance activities on the peer node. Try to use maintenance mode only when necessary.



To enable maintenance mode using Splunk CLI, refer to the following stanza.

```
splunk enable maintenance-mode
```

To return the standard bucket-rolling behavior to normal mode, refer to the following stanza.

```
splunk disable maintenance-mode
```

## Rolling Restart in Splunk Using Splunk CLI

Rolling restart performs a phased restart of all peer nodes. The rolling restart ensures that load-balanced forwarders sending data to the cluster always have a peer available to receive the data. You can perform a rolling restart on all nodes at the same time but it is not recommended. To perform a rolling restart based on an approximate percentage, the master generally tells the number of restart slots to keep open during the rolling restart process.

### Specify the Percentage of Peer to Restart at a Time Using Splunk CLI

The restart percentage is configurable by using Splunk Web, a configuration file or CLI. To Specify the Percentage of Peer to Restart using Splunk CLI refer to the command provided below.

```
splunk edit cluster-config -percent_peers_to_restart <percentage>
```

### Searchable Rolling Restart Using Splunk CLI

Splunk Enterprise 7.1.0 and later provides a searchable option for rolling restarts. The searchable option lets you perform a rolling restart of peer nodes with minimal interruption of ongoing searches. You can use a searchable rolling restart to minimize search disruption when a rolling restart is required due to regular maintenance or a configuration bundle push. To perform a searchable rolling restart, refer to the following stanza on the master node.

```
splunk rolling-restart cluster-peers -searchable true
```

If you want to proceed with the searchable rolling restart despite the health check failure, use the force option on the master node. It is not advisable because it may lead to clearing queue data that is not indexed by the indexer. Refer to the following stanza.

```
splunk rolling-restart cluster-peers -searchable true \
-force true \
-restart_inactivity_timeout <secs> \
-decommission_force_timeout <secs>
```

## Remove Excess Buckets Copies from the Indexer Cluster

Excess bucket copies are more than what's required in the cluster's replication factor or search factor. If the replication factor in the master node is 2 and you have three copies, you are exceeding bucket copies. Remove the excess buckets copies from the index. Excess bucket copies originate when planning for one peer going down for maintenance. So, the master node may initiate bucket fixing activities to make a bucket available for searching, and when the peer node is back from maintenance, excess bucket copies are created.

To remove extra bucket copies from a Splunk indexer, refer to the following command.

```
splunk remove excess-buckets <[index_name]>
```

## Remove a Peer from Master's List

After a peer goes down permanently, it remains on master node lists. For example, a peer goes down permanently, but it continues to appear on the master dashboard although its status changed to Down or Graceful Shutdown, depending on how it went down. In these cases, you need to remove the peer permanently from the master node.

To remove a peer from the master list using Splunk CLI, refer to the following stanza.

```
splunk remove cluster-peers -peers <guid>,<guid>,<guid>,...
```

For more information, go to <https://docs.splunk.com/Documentation/Splunk/7.3.0/Indexer/Aboutindexesandindexers>. Let's now discuss the best strategy for multisite index clusters.

## Managing a Multisite Index Cluster

Managing multisite index clustering in Splunk consists of various tasks, like handling multisite failure, restarting indexing in a multisite cluster after a master restart or site failure, converting a multisite index cluster to a single site, and moving a peer to a new site.

### Master Site in Multisite Index Cluster Fails

If the master node fails in a multisite index clustering, it creates congestion in your Splunk environment. In the meantime, the cluster tries to function normally as much as possible. The peer also continues to stream data to other peers based on the list of target peers that they were using when the master went down, but it's not enough for smooth functioning. So as a best practice, you need to configure a standby server to handle multisite server failure.

### Configure Standby Server

To configure a standby server, you need to take care of the following two things.

- Back up the files that the replacement master needs.
- Ensure that the peer and search head nodes can find the new master.

### Back up the Files That the Replacement Master Needs

There are two static configurations that you must back up to copy it to the replacement master.

- The master's `server.conf` file, where the master's cluster settings are stored.
- The master's `$SPLUNK_HOME/etc/master-apps` directory must be common where all peer node configuration is stored.

## Ensure That the Peer and Search Head Nodes Can Find the New Master

To ensure that the peer and search head nodes can find a new master, you need to follow any one rule from the given rules.

- The replacement must use the same IP address and management port as the primary master.
- If the replacement does not use the same IP address and management port as the primary master, then configure the peer node and add a new `master_uri` address.

## Restart Indexing in the Multisite Cluster After a Master Restart or Site Failure

When a master restarts in Splunk, it blocks data indexing until enough peers have joined the indexer cluster. In a multisite cluster, you might want to restart indexing even though you do not have enough available peers to fulfill all aspects of the site replication factor. Run the Splunk `set indexing-ready` command on the master to unblock indexing when the replication factor of peers is not available.

**Restart** indexing (using Splunk CLI) in the multisite cluster after a master restart or site failure, refer to the following stanza.

```
splunk set indexing-ready -auth admin:your_password
```

## Move a Peer to a New Site

If you want to move a peer node from one site to another in a multi-cluster environment, you can do it using the following instructions.

1. Take the peer offline with the `offline` command. The master reassigns the bucket copies handled by this peer to other peers on the same site.
2. Ship the peer's server to the new site.

3. Delete the entire Splunk Enterprise installation from the server, including its index database and all bucket copies.
4. Reinstall Splunk Enterprise on the server, re-enable clustering, and set the peer's site value to the new site location.

Let's now discuss how to use REST API endpoints in Splunk.

## REST API Endpoints

Splunk REST API endpoints can do almost all operations in Splunk—from authentication to searching to configuration management. The API is divided into endpoints (URIs) served by `splunkd` (i.e., management port 8089). REST API endpoints can be used in Splunk by a programmer for remote querying, searching remotely, and using a third-party to integrate their apps with Splunk. Splunk provides a Software Development Kit (SDK) for programmers to integrate their app with Splunk. The SDK is like a wrapper that calls the REST API and helps abstract the details by providing easy-to-use objects that can interact with Splunk.

In REST API endpoints, you use the open-source command-line tool, `cURL`. There are other command-line tools available, such as `wget`. `cURL` is available on Mac and Linux by default. It can also be downloaded for Windows; go to <http://curl.haxx.se/> for more information.

There are three main methods in Splunk REST API endpoints.

- The GET method gets data that is associated with a resource; for example, accessing search for a result.
- The POST method creates or updates an existing resource.
- The DELETE method deletes a resource.

REST API endpoints have three main functions.

- Running searches
- Managing knowledge objects and configuration
- Updating Splunk Enterprise configuration

## Running Searches Using REST API

In Splunk, REST API endpoints run saved searches. When you want to run a search in Splunk, there are parameters that need to be addressed.

- **max\_count:** Set this parameter if the search result is greater than 10,000 events.
- **status\_buckets:** To access a summary and timeline information from a search job, specify a value for this parameter.
- **rf:** Use this parameter to add a parameter in Splunk.

### Create a Search Job

To create a search job in Splunk, follow these instructions.

1. Open the terminal and browse to \$SPLUNK\_HOME/bin.
2. Execute the following curl command.

```
curl -u admin:changeme -k https://localhost:8089/services/
search/jobs -d search="<search string>"
```

3. In Search String, provide your search command.
4. Splunk returns XML with a search ID (SID). By default, SID is valid for 10 minutes. The following code block is the reply from Splunk.

```
<?xml version='1.0' encoding='UTF-8'?>
<response>
  <sid>1568421821.56</sid>
</response>
```

5. To check the status of your search, type the following command.

```
curl -u admin:changeme -k https://localhost:8089/services/
search/jobs/<SID>
```

If you want to know whether your search was successful or not, in reply, you get a message **Job Done**.

- To get the results of your search operation, execute the following command where you need to just replace Search ID with your Search ID.

```
curl -u admin:changeme -k https://localhost:8089/services/
search/jobs/<SID>/results
```

- To get the result in CSV or JSON format, refer to the following command.

```
curl -u admin:changeme -k https://localhost:8089/services/
search/jobs/<SID>/results --get -d output_mode=csv
```

## Manage Configurations File in Splunk

In Splunk using the REST API, there are two sets of endpoints that provide access to the configuration files: `properties/` and `configs/conf-(file)/`.

- properties/** edits the configuration file in Splunk using REST API endpoints.
- configs/conf-(file)/** is used for setting permission, moving a resource, and so forth.
- properties endpoints** offer various options for listing configurations. GET operations are available to drill down from the list of configuration files to the key/value pairs. Note the following example.

```
curl -k -u admin:pass https://localhost:8089/servicesNS/nobody/
search/properties/props
```

- configs/conf-(file)/ endpoints** use the POST operation to add a stanza to the named configuration file. You can also specify key/value pairs for the newly added stanza. Note the following example.

```
curl -k -u admin:pass https://localhost:8089/servicesNS/nobody/
search/configs/conf-props \
    -d name=test89 \
    -d SHOULD_LINEMERGE=false
    -d ..... \
```

This brings us to the end of Splunk REST API endpoints. Now let's move to the Splunk SDK.

## Splunk SDK

In this section, you learn about using Splunk's Software Development Toolkit (SDK), which is available in Python, Java, JavaScript, and C#. Generally, since the Python SDK is used most often, it is covered in this book. Using the Software Development Toolkit, you can perform the following operations in Splunk.

- Run external searches in Splunk using the REST API
- Create dashboards, advanced models for machine learning, and visualization using other applications
- Send data directly to Splunk using REST API endpoints
- Extract data from Splunk to preserve it

## Python Software Development Kit for Splunk

The Splunk SDK for Python helps programmers interact with Splunk for various operations, including searching, saved searches, data input, REST API endpoints, building applications, and so forth. The Splunk SDK for Python has been tested on Python versions 2.7 and 3.5. To work with the Python SDK for Splunk, you need to set the PYTHONPATH environment variable to Splunk SDK. To install Splunk in Python, go to <https://github.com/splunk/splunk-sdk-python>.

## Program for Data Input in Splunk Using splunklib.client

To work with an SDK using splunklib.client, you need not configure the .splunkrc file; have the splunklib.client file in your Python environment. The following is a simple program showing how to push data from a particular file to the main index in your Splunk environment.

```
import splunklib.client as client

HOST = "localhost"
PORT = 8089
USERNAME = "admin"
PASSWORD = "Deep@1234"
```



```

service = client.connect(
    host=HOST,
    port=PORT,
    username=USERNAME,
    password=PASSWORD)

myindex = service.indexes["main"]
file=open("/Applications/Splunk/copy.txt", "w")
file.write("hi")
file.close()
uploadme = "/Applications/Splunk/copy.txt"
myindex.upload(uploadme);
print(myindex.upload(uploadme))
print("successful")

```

After executing the program, go to your Splunk Web environment and check out the output. It is similar to Figure 16-1.

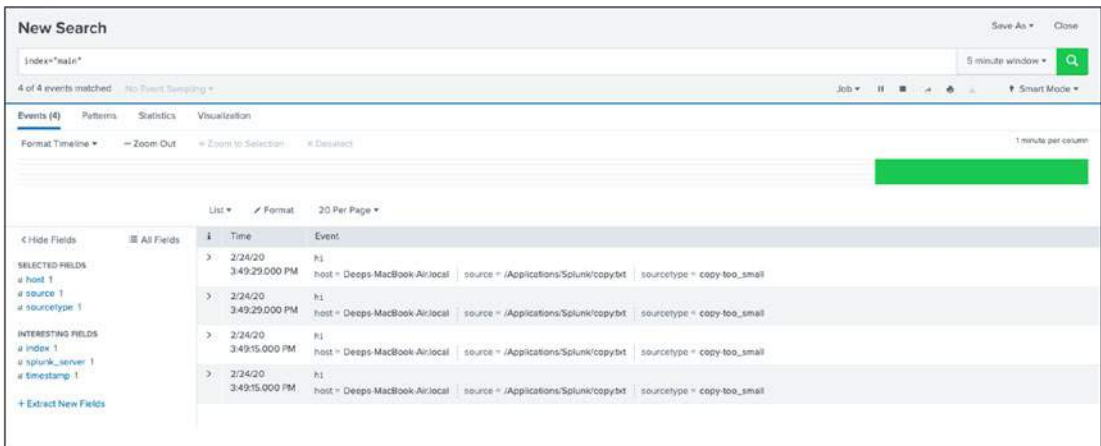


Figure 16-1. Output “Search.py” Events using Splunk Web

## Program for Search in Splunk Using a Command Line

To work with the examples provided within the Splunk SDK, you need to configure the .splunkrc file in your respective environment at <http://dev.splunk.com/view/python-sdk/SP-CAAEEFC>.

The following is the `.splunkrc` file configuration syntax.

```
host=<ip>,<hostname>,<localhost>
port=<mgmt port>
username=<username>
password=<symetry key>
```

- **host** is the address on which you have access to the Splunk instance
- **port** refers to your `mgmt_port` (8089 by default)
- **username** is the username of Splunk instance (admin by default)
- **password** secures your account

The following is the `.splunkrc` file.

```
host=localhost
port=8089
username=admin
password=Deep@1234
```

After configuring the `.splunkrc` file, try to run a `search.py` program in Python using a command prompt (Windows) or terminal (Linux). Go to the example `splunk-sdk` folder and run the following command.

```
python search.py " search index=main|head 1"
```

The output of the `search.py` program is shown in Figure 16-2.

```

[(base) bash-3.2$ python search.py " search index=main|head 1"
<?xml version='1.0' encoding='UTF-8'?>
<results preview='0'>
  <meta>
    <fieldOrder>
      <field>_bkt</field>
      <field>_cd</field>
      <field>_indextime</field>
      <field>_raw</field>
      <field>_serial</field>
      <field>_si</field>
      <field>_sourcetype</field>
      <field>_time</field>
      <field>host</field>
      <field>index</field>
      <field>linecount</field>
      <field>source</field>
      <field>sourcetype</field>
      <field>splunk_server</field>
    </fieldOrder>
  </meta>
  <result offset='0'>
    <field k='_bkt'>
      <value><text>main-0-EB365685-88B0-4299-BE30-24A1982CB10C</text></value>
    </field>
    <field k='_cd'>
      <value><text>0:12</text></value>
    </field>
    <field k='_indextime'>
      <value><text>1582541220</text></value>
    </field>
    <field k='_raw'><v xml:space='preserve' trunc='0'>hi</v></field>
    <field k='_serial'>
      <value><text>0</text></value>
    </field>
    <field k='_si'>
      <value><text>Deeps-MacBook-Air.local</text></value>
      <value><text>main</text></value>
    </field>
    <field k='_sourcetype'>
      <value><text>copy-too_small</text></value>
    </field>
    <field k='_time'>
      <value><text>2020-02-24T16:17:00.000+05:30</text></value>
    </field>
    <field k='host'>
      <value><text>Deeps-MacBook-Air.local</text></value>
    </field>
    <field k='index'>
      <value><text>main</text></value>
    </field>
    <field k='linecount'>
      <value><text>1</text></value>
    </field>
    <field k='source'>
      <value><text>/Applications/Splunk/copy.txt</text></value>
    </field>
    <field k='sourcetype'>
      <value><text>copy-too_small</text></value>
    </field>
    <field k='splunk_server'>
      <value><text>Deeps-MacBook-Air.local</text></value>
  </result>
</results>

```

**Figure 16-2.** Output “search.py” Events using Terminal

You can run other example .py files and go through the reference material to become more familiar with the Splunk SDK and with the Splunk environment.

## Summary

In this journey of indexes, you learned how to configure custom indexes, remove indexes, index data, and configure index parallelization. You saw how to move index database, configure maximum index size, and set limits for disk usage. You learned how to configure a node to be in an offline state, enable maintenance mode, roll restart, remove bucket copies, and remove peer nodes. You learned about Splunk REST API endpoints, the Splunk SDK, and the Python SDK and executed a test program.

This chapter covered a large portion of Splunk architect exam blueprint. You are familiar with the 5% of Module 7 (performance monitoring and tuning), 7% of Module 17 (indexer cluster management and administration) 5% of Module 15 (single-site index cluster), 5% of Module 16 (multisite index clusters), and part of Module 20 (KV Store collection and lookup management).

This brings us to the end of our wonderful journey. I wish you all the best and happy Splunking!!!

## Multiple-Choice Questions

- A. A Splunk admin wants to configure pipeline sets for index parallelization. Select the correct option from the following.
1. [general]  
parallelIngestionPipellines = 2
  2. [general]  
parallelIngestionPippellines = 2
  3. [general]  
parallelIngestionPipelines = 2
  4. [general]  
parallelIngestionPiipellines = 2

- B. The Splunk Software Development Kit is available in which languages? (Select all options that apply.)
1. Python
  2. Java
  3. JavaScript
  4. Impala
  5. Scala
  6. none of the above
- C. Which methods are for REST API endpoints in Splunk?
1. GET
  2. POST
  3. DELETE
  4. INSERT
  5. none of the above
- D. Which is the default management port in Splunk?
1. 8000
  2. 8089
  3. 8001
  4. 8098
- E. To configure event indexes, which file do you need to edit?
1. input.conf
  2. deployment.conf
  3. output.conf
  4. indexes.conf

**Answers**

- A. 3
- B. 1, 2, 3
- C. 1, 2, 3
- D. 2
- E. 4

**References**

- <https://github.com/splunk/splunk-sdk-python>
- <https://dev.splunk.com/enterprise/docs/python/sdk-python/examplespython/commandline>
- <https://docs.splunk.com/DocumentationStatic/PythonSDK/1.6.5/client.html>
- <https://docs.splunk.com/Documentation/Splunk/latest/RESTREF/RESTlist>
- Splunk 7X Quick Start Guide
- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Indexer/Setupmultipleindexes>
- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Indexer/RemovedatafromSplunk>
- <https://docs.splunk.com/Documentation/Splunk/8.0.2/Indexer/Pipelinesets>

## CHAPTER 17

# Final Practice Set

This chapter features multiple-choice questions that are useful for Splunk admin and architect certification exams. You get a better idea of the types of questions that appear on these exams.

## Questions

- A. **In a four-site indexer cluster, which configuration stores two searchable copies at the origin site, one searchable copy at site2, and a total of four searchable copies?**
1. `site_search_factor = origin:2, site1:2, total:4`
  2. `site_search_factor = origin:2, site2:1, total:4`
  3. `site_replication_factor = origin:2, site1:2, total:4`
  4. `site_replication_factor = origin:2, site2:1, total:4`
- B. **If you suspect that there is a problem interpreting a regular expression in a monitor stanza, which log file would you search to verify?**
1. `btool.log`
  2. `metrics.log`
  3. `splunkd.log`
  4. `tailing_processor.log`
- C. **When should multiple search pipelines be enabled?**
1. Only if disk IOPS is at 800 or better.
  2. Only if there are fewer than 12 concurrent users.

3. Only if running Splunk Enterprise version 6.6 or later.
  4. Only if CPU and memory resources are significantly under-utilized.
- D. **Which tool(s) can be leveraged to diagnose connection problems between an indexer and forwarder? (Select all that apply.)**
1. Telnet
  2. tcpdump
  3. Splunk btool
  4. Splunk btprobe
- E. **Which CLI command converts a Splunk instance to a license slave?**
1. splunk add licenses
  2. splunk list licenser-slaves
  3. splunk edit licenser-localslave
  4. splunk list licenser-localslave
- F. **Which Splunk server role regulates the functioning of an indexer cluster?**
1. indexer
  2. deployer
  3. master node
  4. Monitoring Console
- G. **To improve Splunk performance, the `parallelIngestionPipelines` setting can be adjusted on which of the following components in the Splunk architecture? (Select all that apply.)**
1. indexers
  2. forwarders



3. search head
  4. cluster master
- H. **Of the following types of files within an index bucket, which file type may consume the most disk?**
1. raw data
  2. Bloom filter
  3. metadata (.data)
  4. inverted index (.tsidx)
- I. **Which server.conf attribute should be added to the master node's server.conf file when decommissioning a site in an indexer cluster?**
1. site\_mappings
  2. available\_sites
  3. site\_search\_factor
  4. site\_replication\_factor
- J. **Which two sections can be expanded using the search job inspector?**
1. execution costs
  2. saved search history
  3. search job properties
  4. optimization suggestions
- K. **What does the deployer do in a search head cluster (SHC)? (Select all that apply.)**
1. Distributes apps to SHC members
  2. Bootstraps a clean Splunk install for a SHC
  3. Distributes non-search related and manual configuration file changes
  4. Distributes runtime knowledge object changes made by users across the SHC

- L. **When Splunk indexes data in a non-clustered environment, what kind of files does it create by default?**
  - 1. Index and .tsidx files
  - 2. Raw data and index files
  - 3. Compressed and .tsidx files
  - 4. Compressed and meta data files
- M. **To activate replication for an index in an indexer cluster, what attribute must be configured in indexes.conf on all peer nodes?**
  - 1. repFactor = 0
  - 2. replicate = 0
  - 3. repFactor = auto
  - 4. replicate = auto
- N. **Which search head cluster component is responsible for pushing knowledge bundles to search peers, replicating configuration changes to search head cluster members, and scheduling jobs across the search head cluster?**
  - 1. master
  - 2. captain
  - 3. deployer
  - 4. deployment server
- O. **Which Splunk internal index contains license-related events?**
  - 1. \_audit
  - 2. \_license
  - 3. \_internal
  - 4. \_introspection

- P. What is the default log size for Splunk internal logs?**
1. 10 MB
  2. 20 MB
  3. 25 MB
  4. 30 MB
- Q. When Splunk is installed, where are the internal indexes stored by default?**
1. SPLUNK\_HOME/bin
  2. SPLUNK\_HOME/var/lib
  3. SPLUNK\_HOME/var/run
  4. SPLUNK\_HOME/etc/system/default
- R. In which phase of the Splunk Enterprise data pipeline are indexed extraction configurations processed?**
1. input
  2. search
  3. parsing
  4. indexing
- S. When planning a search head cluster, which of the following is true?**
1. All search heads must use the same operating system.
  2. All search heads must be members of the cluster (no standalone search heads).
  3. The search head captain must be assigned to the largest search head in the cluster.
  4. All indexers must belong to the underlying indexer cluster (no standalone indexers).

**Answers**

- A. 4
- B. 3
- C. 4
- D. 2,3
- E. 3
- F. 3
- G. 1,2
- H. 2
- I. 1
- J. 2,3
- K. 1
- L. 2
- M. 3
- N. 2
- O. 3
- P. 3
- Q. 2
- R. 3
- S. 3

## Summary

This chapter featured multiple-choice questions useful for admin and architect certification. You have come to the end of Module 3. The next chapter sets up a Splunk environment on the AWS platform.

## CHAPTER 18

# Setting up a Splunk Environment with AWS

In this chapter, you learn about AWS deployment using Splunk.

This chapter covers the following topics.

- Amazon Web Services
- Configuring a Splunk instance on EC2
- Deploying multisite index clustering
- Deploying search head clustering
- Deploying a configuration file using a cluster master
- Monitoring a distributed environment

## Amazon Web Services

Amazon launched its cloud computing service, Amazon Web Services (AWS), in 2006. Cloud computing can be broken down into infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS).

IaaS is a type of cloud service that includes all the components needed to compute, including machines, block storage, routers, and firewalls. IaaS enables end-user scalability based on user requirements, and end users need to pay only for the services used. AWS supports infrastructure and is well known for IaaS. Amazon Virtual Private Cloud, AWS Direct Connect, Amazon Route 53, Amazon CloudFront, Amazon Elastic Load Balancing, Amazon Elastic Cloud Computing, Amazon Elastic Container Services, and Amazon Lambda are Amazon's IaaS services.

PaaS is a type of cloud service that includes the hardware and software tools needed for application development. PaaS includes highly available and relational NoSQL databases, big data ingestion, and scalable web-accessible storage. AWS has services that support PaaS, including Amazon Elastic Block Storage, Amazon Elastic File System, Amazon Simple Storage Service, Amazon Glacier, and AWS Storage Gateway.

SaaS is a type of cloud service in which users directly consume services. End users do not need to understand the workings of the services they want to use. The AWS Partner Network provides SaaS to clients, including Amazon WorkMail, Amazon WorkDocs, and Amazon WorkSpaces.

In this chapter, you use only Amazon Elastic Compute Cloud (EC2) instances to set up Splunk for AWS.

## Configuring an EC2 Instance Using the AWS Management Console

Amazon EC2 instances can be configured in many ways, but the easiest and best way is to go to <https://aws.amazon.com/> and sign in to the console. Once you have signed in to your AWS account, you gain access to the Amazon Management Console page. You can manage your entire account from this page. Let's have a look at it (see Figure 18-1).

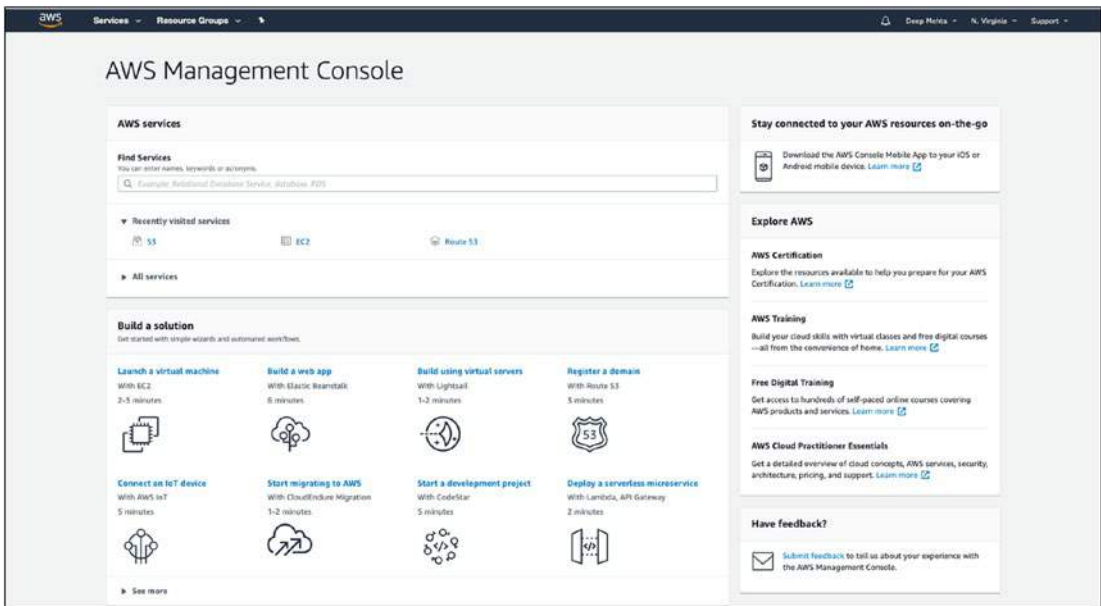


Figure 18-1. AWS Management Console

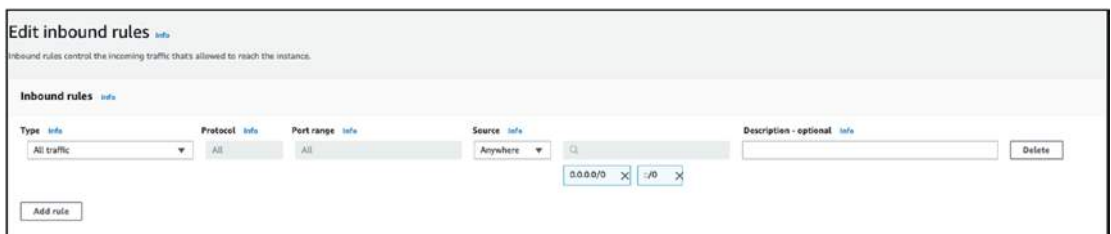
In this section, you use only EC2 instances to configure Splunk instances. In AWS, search for EC2 instances and then go to the EC2 page.

1. On the left side of the EC2 instance, there is the Network & Security menu bar.
2. Go to Key Pairs.
3. Add a key (I created a key named *test*.) Refer to Figure 18-2.



**Figure 18-2.** Key Pair:Test

4. In Network & Security, go to Security Groups.
5. Create a new security group. (I created a security group named Main).
6. Edit Inbound rules where in traffic select “All traffic”, protocol select “All”, Port Select “All” and Destination Source “Anywhere.” Refer to Figure 18-3.

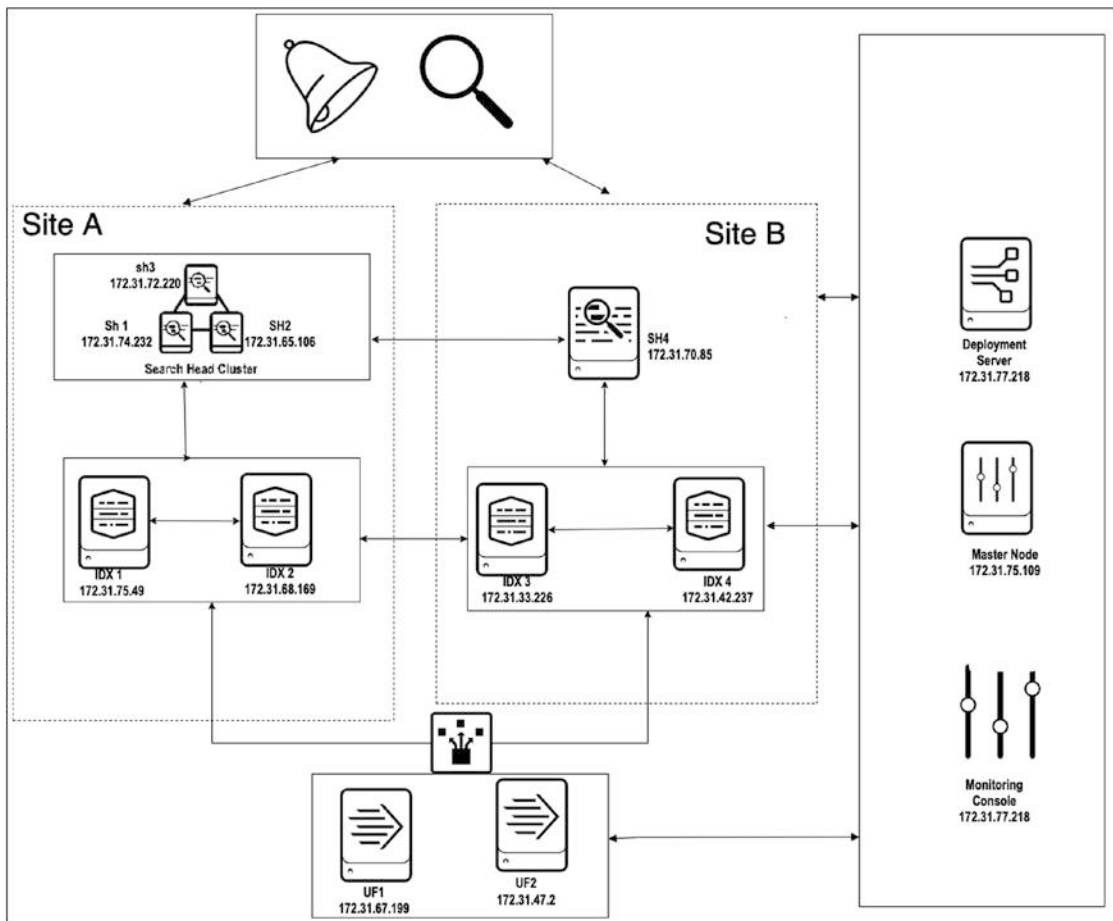


**Figure 18-3.** Security Group:Main

7. Click the Launch instance in EC2 dashboards.
8. Select an image (whichever you want). I selected Amazon Linux 2 AMI for this exercise. Click Next. Instance type provides a wide selection of instance type optimized for use cases (I selected t2.micro for this exercise).

9. Click Review and Launch.
10. Click Edit Security Groups ► Existing Security Groups, and then select the group you have created (Main for me).
11. Click Launch and select the key pair that you created. (I created a key name Test.)

In this chapter, you deploy Splunk multisite clustering for site A and site B. Each site has two indexers, search head clustering, a license master, a cluster master, and a deployment server/Monitoring Console on the same instance, and there are two universal forwarders. The public IP of the instance is also provided. Refer to the diagram shown in Figure 18-4.



**Figure 18-4.** Splunk Deployment Constituent Architecture



To deploy multisite clustering, you need to have 12 different Splunk instances. Create EC2 instances according as warranted. (This is just a test exercise to implement multisite index clustering.) For production deployment, it is not recommended at all. In this exercise we have 5 indexers, 4 search head, 2 universal forwarder and a deployment server/monitoring console. You can refer Figure 18-5 for it.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv4 IPs	Key Name	Monitoring
idx1	i-0c107353d0f1e8d	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-54-172-9-89.comp...	54.172.9.89	-	test	disabled
sh2	i-0b7e8e6f770cedc8f	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-18-208-119-171.co...	18.208.119.171	-	test	disabled
ds1mc	i-0b234731d3dc28ccc	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-54-83-163-129.co...	54.83.163.129	-	test	disabled
idx1	i-0905b17fa135186e	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-3-85-148-2.comput...	3.85.148.2	-	test	disabled
sh1	i-07af8e552c783a7d	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-34-232-66-44.co...	34.232.66.44	-	test	disabled
search_captain	i-07804411623c8a8b	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-34-204-82-176.co...	34.204.82.176	-	test	disabled
idx3	i-06b28161963f9883b	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-54-237-84-204.co...	54.237.84.204	-	test	disabled
idx2	i-061e8f84269c0c58bc	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-52-61-178-113.co...	52.91.178.113	-	test	disabled
license_ma...	i-0510ec7882a0c0b88	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-52-201-228-163.co...	52.201.228.163	-	test	disabled
uf1	i-04c3b8935691c0d2	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-35-174-209-196.co...	35.174.209.196	-	test	disabled
uf2	i-03b4ca256bae131e0	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-34-207-169-159.co...	34.207.169.159	-	test	disabled
cmaster	i-030c75b78a6330e	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-54-210-168-27.co...	54.210.168.27	-	test	disabled

Figure 18-5. Splunk Deployment Components:EC2 instance

## Configuring Splunk on an EC2 Instance

You have created 12 Amazon Web Services EC2 instances for multisite Splunk deployment. In this deployment, you need to configure two Splunk universal forwarders and ten instances with Splunk images.

## Configuring Splunk Enterprise

To configure a Splunk Enterprise instance on Amazon Web Services EC2. Log in to the EC2 instance using your credentials and then refer to the following instructions.

1. Create a directory in /opt named **splunk**.

```
$ sudo mkdir /opt/splunk
```

2. Configure the Splunk Enterprise image using Splunk widget.
3. Extract the Splunk image in /opt. The following is an example.

```
$ sudo tar -xzvC /opt -f <image name>
```

4. Install the Splunk image and accept a Splunk license.

```
$ sudo ./splunk install --accept-license
```

5. Set the username and password of your choice. Start the Splunk server.

```
$ sudo ./splunk start
```

6. Set the Splunk server name and hostname based on the instance type; for example, if it's an index 1 name it idx1.

7. 

```
$ sudo sudo ./splunk set servername <
```

```
$ sudo ./splunk enable boot-start
```

8. Enable the Splunk port of your choice. I enabled port 9997.

```
$ sudo ./splunk enable listen 9997
```

## Configuring Splunk Forwarder

Let's configure a Splunk forwarder on Amazon Web Services EC2 instances. Log in to EC2 instances and refer to the following instructions.

1. Create a directory in /opt named splunk forwarder.

```
$ sudo mkdir /opt/splunk forwarder
```

2. Configure a Splunk forwarder image using Splunk widget.

3. Extract a Splunk image in /opt. The following is an example.

```
$ sudo tar -xzvC /opt -f <image name>
```

4. Install the Splunk image and accept the Splunk license.

```
$ sudo ./splunk install --accept-license
```

5. Set the username and password of your choice. Start the Splunk server.

```
$ sudo ./splunk start
```

6. Set the Splunk server name and Splunk hostname based on instance type; for example, if it's forwarder 1, name it ufl.

```
name>
$ sudo sudo ./splunk set servername <name>
$ sudo ./splunk set default-hostname <name>
Enable Splunk boot start$ sudo ./splunk set default-hostname
<name>
```

7. Enable Splunk boot start.

```
$ sudo ./splunk enable boot-start
```

Next, let's move to Splunk multisite index clustering to configure a master site and slave nodes.

## Deploying Multisite Index Clustering

In this section, you will configure multisite index clustering in Splunk where you have two sites Site 1(A) and Site 2(B) (refer Figure 18-4). In this exercise you will deploy multi site index clustering where replication factor = 1: total= 2 copies and searchable copies = 1: total = copies.

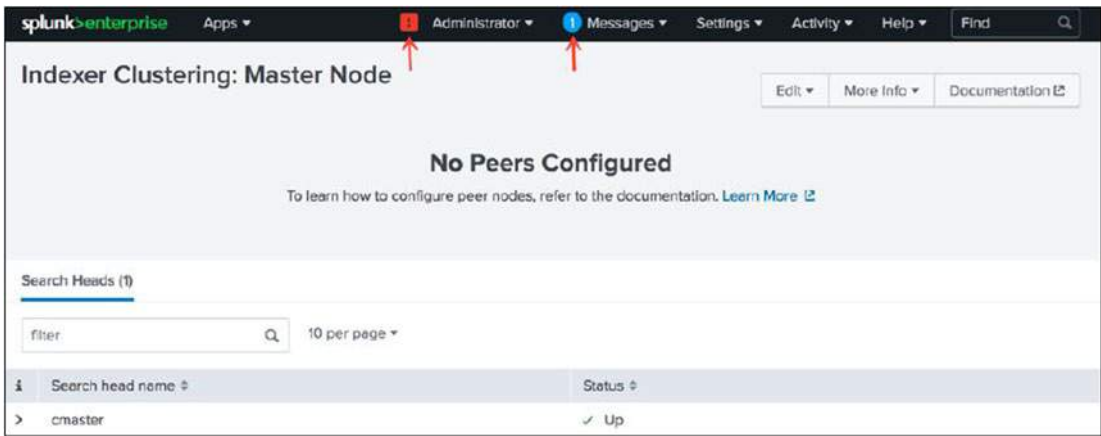
### Configuring a Cluster Master

To configure a cluster master in Splunk, log in to an EC2 instance for a cluster master and go to \$SPLUNK\_HOME/bin/.

To enable a cluster master in Splunk, refer to the following command which will have searchable copies of 1 and replication factor of 1.

```
$ sudo ./splunk edit cluster-config -mode master -multisite true -site site1
-available_sites site1, site2 -site_replication_factor origin:1, total:2
-site_search_factor origin:1, total:2 -replication_factor 1 -search_factor 1
-secret idxcluster
```

To monitor the cluster master's status, navigate to Splunk Web's settings and go to Indexer Clustering, where you find a screen similar to Figure 18-6.



**Figure 18-6.** *Monitoring Indexer Clustering:Master Node*

## Configuring a Slave Node

To configure a slave node for a cluster master in Splunk, log in to the EC2 instance of an indexer and go to \$SPLUNK\_HOME/bin/.

To enable indexer 1 and indexer 2, report to the master node with site 1. Refer to the following command.

```
$ sudo ./splunk edit cluster-config -mode slave -master_uri
https://172.31.75.109:8089 -secret idxcluster -replication_port 9200
-site site1

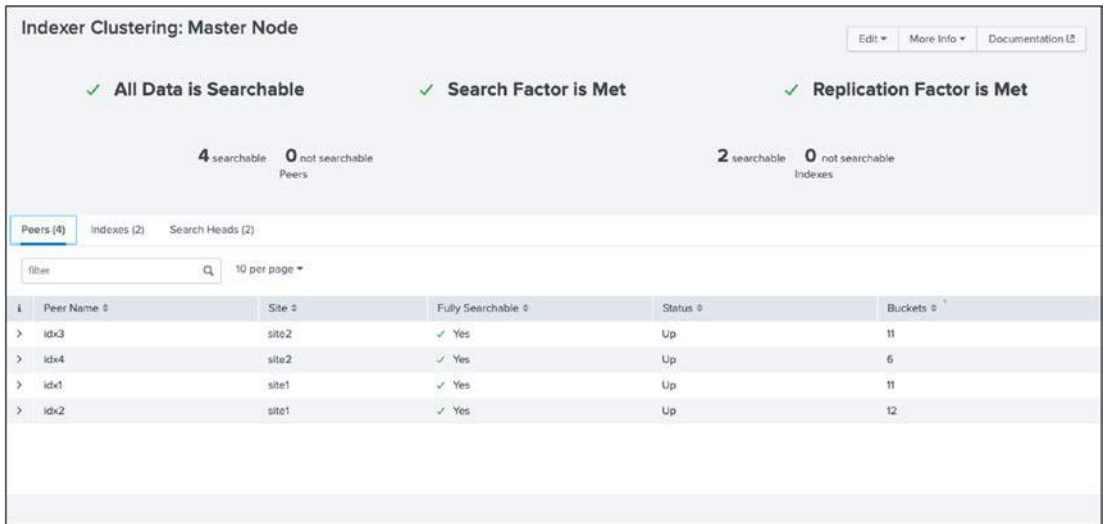
sudo ./splunk restart
```

To enable indexer 3 and indexer 4, report to the master node with site 2. Refer to the following command.

```
$ sudo ./splunk edit cluster-config -mode slave -master_uri
https://172.31.75.109:8089 -secret idxcluster -replication_port 9200
-site site2

sudo ./splunk restart
```

Configure the master node and all indexers to their respective sites. To start monitoring the cluster, move to the cluster master and navigate to Splunk Web’s settings. Go to Indexer Clustering, where you find all the Splunk instances reported to the cluster master. Figure 18-7 shows that idx1, idx2, idx3, and idx4 were reported to the master node.



**Figure 18-7.** Monitoring Indexer Cluster:Peer node

Next, let's move to deploying search head clustering.

## Deploying a Search Head

In this section, you configure multisite search head clustering for site 1 and deploy the search head for site 2.

## Configuring a Search Head

To configure a slave node for a search head in Splunk, log in to the EC2 instance of a search head and go to `$SPLUNK_HOME/bin/`.

To enable sh1, sh2, and sh3 and report to a master node with site 1, refer to following command.

```
sudo ./splunk edit cluster-config -mode searchhead -master_uri
https://172.31.75.109:8089 -site site1 -secret idxcluster

sudo ./splunk restart
```

To enable sh4 and report to a master node with site 2, refer to the following command.

```
sudo ./splunk edit cluster-config -mode searchhead -master_uri https://172.31.75.109:8089 -site site2 -secret idxcluster  
  
sudo ./splunk restart
```

After configuring the search heads to their respective sites now for monitoring cluster master status whether search head has reported to master node or not navigate to Splunk Web settings and go to indexer clustering where you can find search heads bar so navigate to search heads. You can find all the instances that reported to the master node. Figure 18-8 shows that Deep-SHC (sh3), sh1, sh2, sh4, and cmaster reported to the master node.

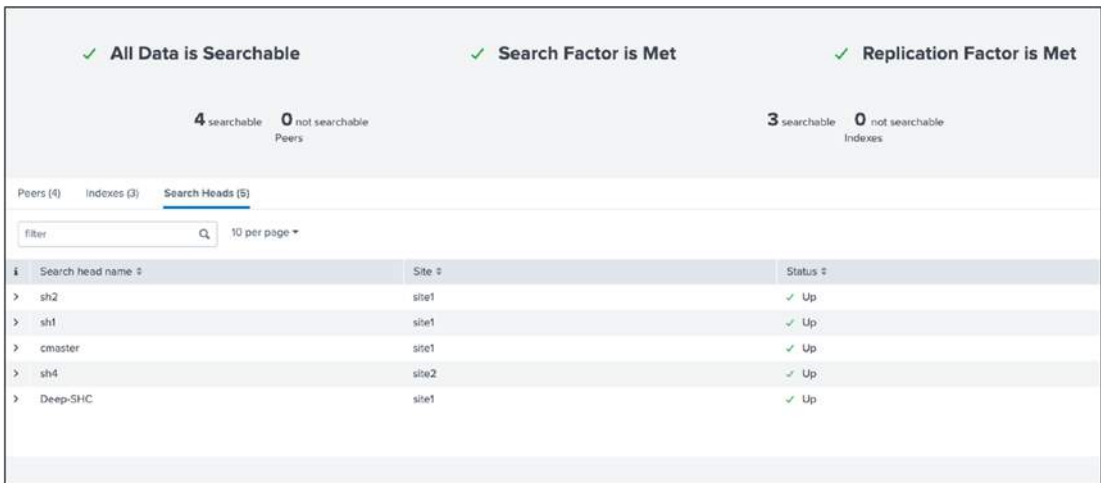


Figure 18-8. Monitoring Indexer Clustering: Search Heads

## Configuring Search Head Clustering

In this section you would configure Splunk search head clustering for site 1 (i.e., sh1, sh2, and sh3 (Deep-shc)). I have a habit of labeling my captain node uniquely so that I can remember it. It is a personal choice. I labeled the sh3 instance as Deep-shc. To enable sh1, sh2, and Deep-shc i.e. sh3 for search head clustering.

To enable sh3 for clustering with a secret key, use the init command. Refer to the following command.

```
sudo ./splunk init shcluster-config -mgmt_uri https://172.31.72.220:8089
-replication_port 9200 -secret shcluster
```

```
Sudo ./splunk restart
```

To enable sh1 for clustering with a secret key, use the init command. Refer to the following command.

```
sudo ./splunk init shcluster-config -mgmt_uri https://172.31.74.232:8089
-replication_port 9200 -secret shcluster
```

```
Sudo ./splunk restart
```

To enable sh2 for clustering with a secret key, use the init command. Refer to the following command.

```
sudo ./splunk init shcluster-config -mgmt_uri https://172.31.65.106:8089
-replication_port 9200 -secret shcluster
```

```
Sudo ./splunk restart
```

Go to the instance where you want to make the node a search head captain; in my case, it's a Deep-shc node. Use the bootstrap command to elect a captain since there is only one node in the cluster. Refer to the following command.

```
sudo ./splunk bootstrap shcluster-captain -servers_list
https://172.31.72.220:8089
```

To add sh1 to the existing Splunk search head cluster, use the add shcluster-member command, but do it from the captain node only. Refer to the following command.

```
sudo ./splunk add shcluster-member -new_member_uri
https://172.31.74.232:8089
```

To add sh2 to the existing Splunk search head cluster, use the add shcluster-member command, but do it from the captain node only. Refer to the following command.

```
sudo ./splunk add shcluster-member -new_member_uri
https://172.31.65.106:8089
```

To check the status of a search head cluster member using a captain, refer to the following command.

```
sudo ./splunk show shcluster-status
```

```
Captain:
    dynamic_captain : 1
    elected_captain : Tue Jun 23 13:16:02 2020
                   id : F5C922BF-95C7-4D41-BDB8-8DDFAC47ABD7
    initialized_flag : 1
                   label : Deep-SHC
                   mgmt_uri : https://172.31.72.220:8089
    min_peers_joined_flag : 1
    rolling_restart_flag : 0
    service_ready_flag : 1

Members:
  sh2
    label : sh2
    last_conf_replication : Tue Jun 23 13:18:30 2020
    mgmt_uri : https://172.31.65.106:8089
    mgmt_uri_alias : https://172.31.65.106:8089
    status : Up

  sh1
    label : sh1
    last_conf_replication : Tue Jun 23 13:18:32 2020
    mgmt_uri : https://172.31.74.232:8089
    mgmt_uri_alias : https://172.31.74.232:8089
    status : Up

  Deep-SHC
    label : Deep-SHC
    mgmt_uri : https://172.31.72.220:8089
    mgmt_uri_alias : https://172.31.72.220:8089
    status : Up
```

**Figure 18-9.** Configuring Search Head Clustering Captain Node

This section implemented multisite search head clustering for site 1 and search head for site 2. In the next section, you use deployment instances to deploy configuration files in Splunk.

## Deploying Configurations

In this section, you would deploy configuration for following task:

- Deploy indexes.conf and props.conf to all indexers using cluster master.
- Deploy shc app to search head clustering using deployment server.



- Configure universal forwarder for indexer discovery.
- Deploy configuration on universalforwarder1 using deployment server to monitor Text.txt file.
- Configure master node and deployment server to send internal logs to indexer.

## Configuring a Cluster Master

The master node can distribute configuration to indexers. The master node deploys an add-on in the middle of the indexing layer. In the use case, a universal forwarder forwards Test.txt data to index test.

1. To deploy a configuration using a master node, migrate to `$SPLUNK_HOME/etc/cmater/_local`.
2. Create or edit the `indexes.conf` file. Refer to the following code section.

```
[test]
homePath=$SPLUNK_DB/test/db
coldPath=$SPLUNK_DB/test/colddb
thawedPath=$SPLUNK_DB/test/thaweddb
repFactor=auto
```

3. Create or edit the `props.conf` file for source type Test9. Refer to the following code section.

```
[Test9]
TIME_PREFIX=\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\s\-\s\d{5}\s+
TIME_FORMAT = %m/%d/%Y %k:%M
MAX_TIMESTAMP_LOOKAHEAD = 15
LINE_BREAKER = ([\r\n]+)\d+\s+"\$EIT\,
SHOULD_LINEMERGE
```

4. To deploy a configuration file to all indexers using a cluster master, go to `$SPLUNK_HOME/bin/`.

```
sudo ./splunk validate cluster-bundle --check restart
```

- To confirm the status of the bundle validation, refer to the following command.

```
sudo ./splunk show cluster-bundle-status
```

If your cluster bundle deploys successfully, you find the checksum of your deployment. The command is similar to Figure 18-10.

```
[[ec2-user@ip-172-31-75-109 bin]$ sudo ./splunk validate cluster-bundle --check restart
Validating new bundle. Please run 'splunk show cluster-bundle-status' to check the status of the bundle validation.
[[ec2-user@ip-172-31-75-109 bin]$ sudo ./splunk show cluster-bundle-status

master
  cluster_status=None
  active_bundle
    checksum=826AF3010CA7165419661216C27A2AAD
    timestamp=1592876799 (in localtime=Tue Jun 23 01:46:39 2020)
  latest_bundle
    checksum=826AF3010CA7165419661216C27A2AAD
    timestamp=1592876799 (in localtime=Tue Jun 23 01:46:39 2020)
  last_validated_bundle
    checksum=826AF3010CA7165419661216C27A2AAD
    last_validation_succeeded=1
    timestamp=1592918946 (in localtime=Tue Jun 23 13:29:06 2020)
  last_check_restart_bundle
    last_check_restart_result=restart not required
    checksum=
    timestamp=0 (in localtime=Thu Jan 1 00:00:00 1970)

idx3  0700F4C5-9908-4036-AE0C-58A347989BAA  site2
  active_bundle=826AF3010CA7165419661216C27A2AAD
  latest_bundle=826AF3010CA7165419661216C27A2AAD
  last_validated_bundle=826AF3010CA7165419661216C27A2AAD
  last_bundle_validation_status=success
  restart_required_apply_bundle=0
  status=Up

idx4  DC29F3C9-32C4-454C-8D53-8CDCAA618549  site2
  active_bundle=826AF3010CA7165419661216C27A2AAD
  latest_bundle=826AF3010CA7165419661216C27A2AAD
  last_validated_bundle=826AF3010CA7165419661216C27A2AAD
  last_bundle_validation_status=success
  restart_required_apply_bundle=0
  status=Up

idx1  DCD95AC2-E49C-4CEB-A9D4-C54800C2B832  site1
  active_bundle=826AF3010CA7165419661216C27A2AAD
  latest_bundle=826AF3010CA7165419661216C27A2AAD
  last_validated_bundle=826AF3010CA7165419661216C27A2AAD
  last_bundle_validation_status=success
  restart_required_apply_bundle=0
  status=Up

idx2  FEB0AD2C-4FEF-43DC-83D1-3230BFCD80EE  site1
  active_bundle=826AF3010CA7165419661216C27A2AAD
  latest_bundle=826AF3010CA7165419661216C27A2AAD
  last_validated_bundle=826AF3010CA7165419661216C27A2AAD
  last_bundle_validation_status=success
  restart_required_apply_bundle=0
  status=Up
```

Figure 18-10. Deploying cluster bundle on peer indexer

## Deploying an App to Search Head Cluster Using Deployment Server

In this section, as provided in the use case, to make a scalable platform, you need not push all configuration by going physically to the node, so you use the deployment server to push configurations. Similarly, you push the shc app to all nodes of search head clustering using the deployment server. The shc app sends internal logs to indexers.

To deploy the shc app using a deployment server, go to `$SPLUNK_HOME/bin/` for search head 1 (sh1). Refer to the following command.

```
sudo ./splunk edit shcluster-config -conf_deploy_fetch_url
https://172.31.77.218:8089
```

To deploy the shc app using a deployment server, go to `$SPLUNK_HOME/bin/` for search head 2 (sh2). Refer to the following command.

```
sudo ./splunk edit shcluster-config -conf_deploy_fetch_url
https://172.31.77.218:8089
```

To deploy the shc app using a deployment server, go to `$SPLUNK_HOME/bin/` for search head 3 (sh3). Refer to the following command.

```
sudo ./splunk edit shcluster-config -conf_deploy_fetch_url
https://172.31.77.218:8089
sudo ./splunk edit shcluster-config -conf_deploy_fetch_url
https://172.31.77.218:8089
```

In the deployment server, create an app named shc located in `$SPLUNK_HOME/etc/shcluster/apps/`.

1. Create or edit `apps.conf` located in `/local/apps.conf` for the shc app. Refer to the following code.

```
[ui]
is_visible = 0
[package]
id = shc_base
check_for_updates = 0
```

2. Create or edit `outputs.conf` located in `/local/outputs.conf` for the `shc` app. Refer to the following code.

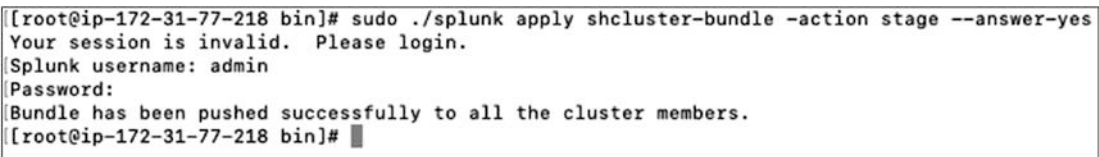
```
[indexAndForward]
index = false

[tcpout]
defaultGroup = default-autolb-group
forwardedindex.filter.disable = true
indexAndForward = false

[tcpout:default-autolb-group]
server=172.31.75.49:9997,172.31.68.169:9997,172.31.33.226:9997,
172.31.42.237:999
7
```

3. To push the configuration in search head clustering nodes, go to `$SPLUNK_HOME/bin/` in the deployment server. Refer to the following command.

```
sudo ./splunk apply shcluster-bundle -action stage --answer-yes
```



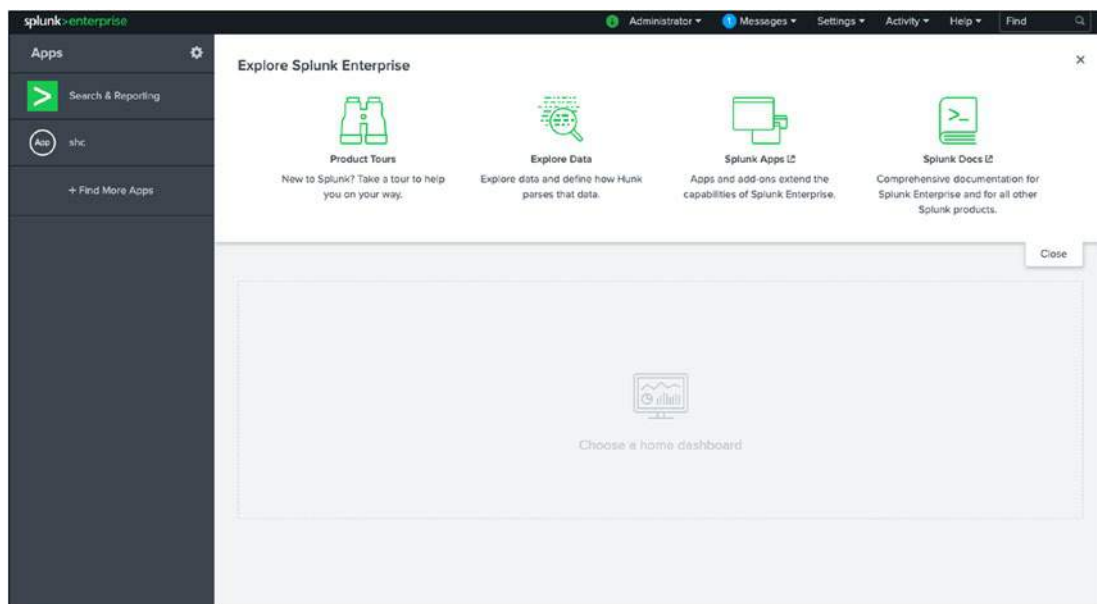
**Figure 18-11.** Deploying cluster bundle on Search Head Clustering

4. The deployment is ready to be pushed, but it is not yet pushed. To push deployment, you need to contact the search head cluster captain and use the `apply shcluster-bundle` command. Refer to the following command.

```
sudo ./splunk apply shcluster-bundle -action send -target
https://172.31.72.220:8089 --answer-yes
```

```
[root@ip-172-31-77-218 bin]# sudo ./splunk apply shcluster-bundle -action stage --answer-yes
Your session is invalid. Please login.
Splunk username: admin
Password:
Bundle has been pushed successfully to all the cluster members.
[root@ip-172-31-77-218 bin]# sudo ./splunk apply shcluster-bundle -action send -target https://172.31.72.220:8089 --answer-yes
Bundle has been pushed successfully to all the cluster members.
[root@ip-172-31-77-218 bin]# █
```

**Figure 18-12.** Deploying cluster bundle on Search Head using Search Head Cluster Captain



**Figure 18-13.** Search Head:SHC App Deployed

## Configuring a Universal Forwarder for Indexer Discovery

In this section, you enable indexer discovery on universal forwarders and configure site failover. Indexer discovery is configured on autoLBFrequency.

## Configuring a Cluster Master for Indexer Discovery

To configure indexer discovery in Splunk deployment, you need to edit `server.conf`.

1. Create or edit `server.conf` located in `$SPLUNK_HOME/etc/system/local/`. Refer to the following code.

```
[indexer_discovery]
pass4SymmKey = my_secret
```

- Restart the Splunk master node by going to `$SPLUNK_HOME/bin/`. Refer to the following command.

```
sudo ./splunk restart
```

## Configuring a Universal Forwarder for Indexer Discovery

To configure indexer discovery in a universal forwarder, you need to edit `outputs.conf`.

- Create or edit `outputs.conf` located in `$SPLUNK_HOME/etc/system/local/` of the forwarder. Refer to the following code.

```
[indexer_discovery:master1]
pass4SymmKey = my_secret
master_uri = https://172.31.75.109:8089

[tcput:group1]
autoLBfrequency = 30
forceTimebasedAutoLB = true
indexerDiscovery = master1
useACK=true
```

- Create or edit `server.conf` located in `$SPLUNK_HOME/etc/system/local/` of forwarder 1. Refer to the following code.

```
[general]
site = site1

[clustering]
forwarder_site_failover = site1:site2
```

- Create or edit `server.conf` located in `$SPLUNK_HOME/etc/system/local/` of forwarder 2. Refer to the following code.

```
[general]
site = site2

[clustering]
forwarder_site_failover = site2:site1
```

4. Restart the Splunk forwarder by going to `$SPLUNK_HOME/bin/`. Refer to the following command.

```
sudo ./splunk restart
```

## Configuring a Universal Forwarder for Indexer Deployment

In this section, you deploy an app named `uf1` on universal forwarder 1 using a deployment server that monitors the `Test.txt` file.

### Configuring a Deployment Server for Deployment

Configure the deployment server for `uf1` app by traversing to the EC2 instance of the deployment server and migrating to `$SPLUNK_HOME/etc/deployment-apps/`.

1. Create an app named `uf1`.
2. Migrate to `uf1/local` and create `inputs.conf` to monitor the `Test.txt` file. Refer to the following stanza.

```
[monitor:///opt/Test.txt]
disabled=false
index=test
sourcetype=Test9
```

### Configuring a Forwarder for the Deployment Server

To configure forwarders for the deployment server, go to the EC2 instance of the forwarders and migrate to `$SPLUNK_HOME/bin/`.

To configure the forwarders to report to the deployment server, refer to the following command.

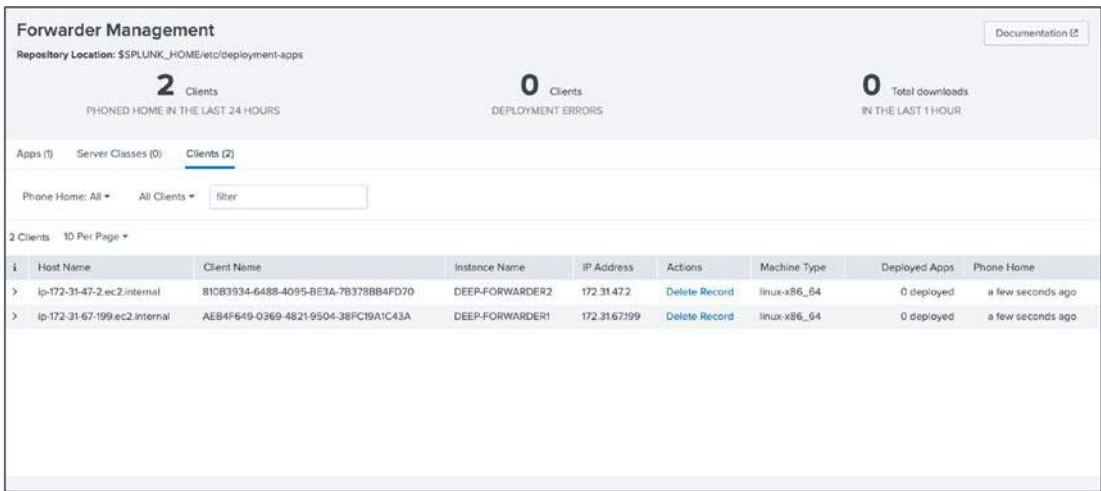
```
./sudo splunk set deploy-poll 172.31.77.218:8089
```

## Deploying the UF1 App

Use forwarder management to deploy the uf1 app on a universal forwarder.

1. Go to deployment server on Splunk Web.
2. Go to Settings in Splunk Web and then to Forwarder Management.

There are two universal forwarders reporting to forwarder management. Refer to Figure 18-14.

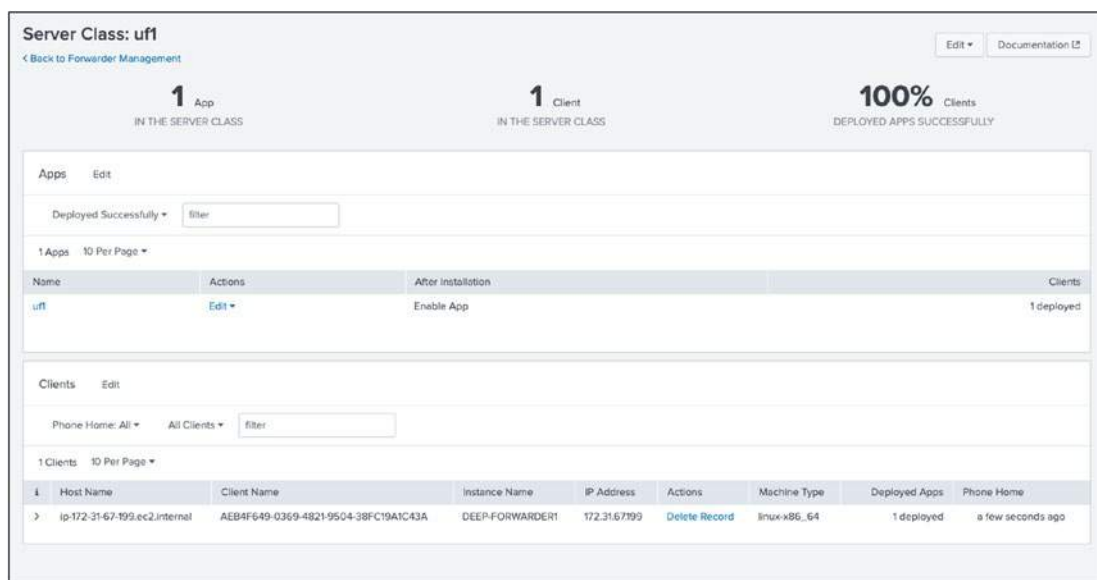


**Figure 18-14.** Forwarder Management: Universal Forwarders

3. Once universal forwarder 1 reports to the deployment server, you need to create a server class. Select the uf1 app to deploy on universal forwarder 1.

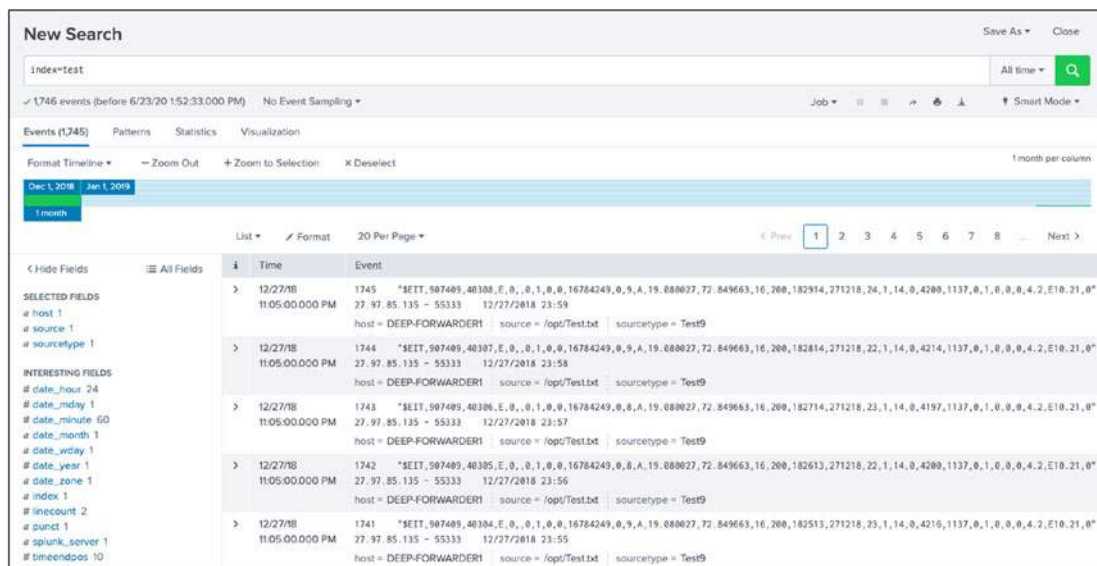
Figure 18-15 shows the uf1 app deployed on universal forwarder 1.





**Figure 18-15.** Server Class:ufl

Once the ufl app is deployed in the universal forwarder. When you log in to our search head 3, and when you type `index="test"` in the search bar, you can observe our data is parsed from `host="DEEP-FORWARDER1"`. Refer to Figure 18-16.



**Figure 18-16.** Events in Index test:DEEP-FORWARDER1

To Confirm whether repFactor=auto for the test index, navigate to the Splunk Web settings on cluster master and go to Indexer Clustering. You can find the status of the test index. It should be searchable and should have replicated data copies. Refer to Figure 18-17.

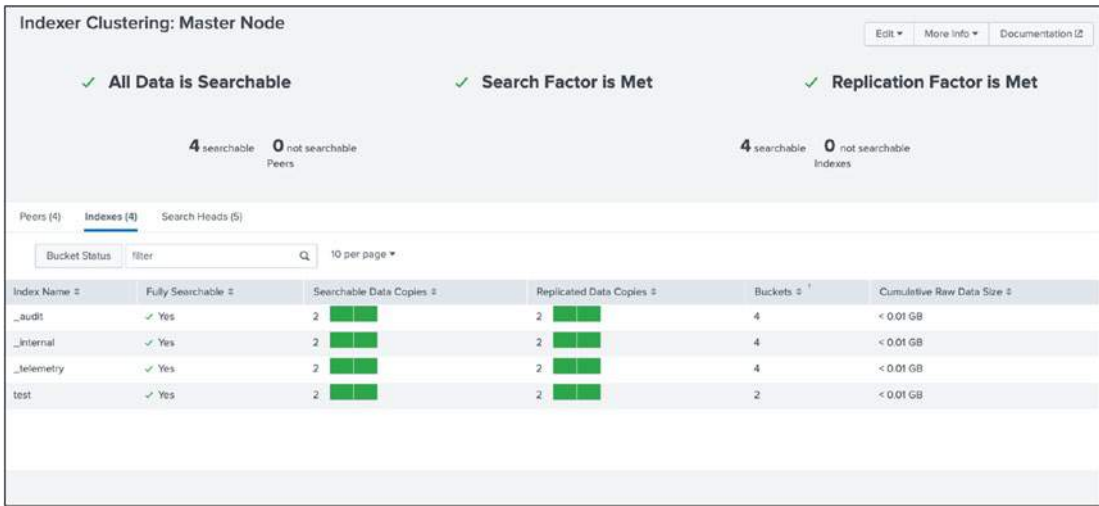


Figure 18-17. Indexer Clustering:test index

## Configuring the Master Node, Deployment Server, and Search Head 4 to Send Internal Logs

In this section, you send internal logs from the deployment server, master node, and search head 4 to indexer 1, indexer2, indexer 3, and indexer 4. Internal logs monitor Splunk Enterprise deployment.

1. Go to the deployment server, cluster master, and search head 4 instances.
2. Go to \$SPLUNK\_HOME/etc/system/local/outputs.conf. Refer to the following code. (For example, replace server=172.31.49.45:9997 with your indexer IP.)

```
[indexAndForward]
index = false
```

```
[tcpout]
defaultGroup = default-autolb-group
forwardedindex.filter.disable = true
indexAndForward = false

[tcpout:default-autolb-group]
server=172.31.75.49:9997,172.31.68.169:9997,172.31.33.226:9997,
172.31.42.237:999
7
```

3. Go to `$(SPLUNK_HOME)/bin/` and type the following command.

```
./sudo splunk restart
```

Now let's look at monitoring distributed environments.

## Monitoring Distributed Environments

In this section, you use a deployment server/Monitoring Console to monitor distributed environments. The Monitoring Console monitors distributed environments. It troubleshoots when a Splunk instance fails or when any other issue occurs in your Splunk environment.

## Adding a Search Peer to Monitor

In this section, you monitor your Splunk environment using a deployment server/Monitoring Console in Amazon Web Services EC2 instances. To monitor the Splunk environment, you need to add a search peer on a cluster master and search heads. To add a search peer, refer to the following instructions.

1. Go to Monitoring Console Instance and Move to settings ---> distributed search using splunk web.
2. Click New and enter the peer URI followed by the remote username and remote password. Confirm the password.
3. Click Save.

Once you have added all the search peers to your instance, you see a screen similar to Figure 18-18.

Peer URI	Splunk instance name	State	Replication status	Cluster label	Health status	Health check failures	Status	Actions
172.31.33.226:8089	idx3	Up	Initial	5212C962-77E5-416B-8BE3-CE030286C451	Healthy	None	Enabled   Disable	Quarantine   Delete
172.31.42.237:8089	idx4	Up	Initial	5212C962-77E5-416B-8BE3-CE030286C451	Healthy	None	Enabled   Disable	Quarantine   Delete
172.31.65.106:8089	sh2	Up	Initial	5212C962-77E5-416B-8BE3-CE030286C451	Healthy	None	Enabled   Disable	Quarantine   Delete
172.31.68.369:8089	idx2	Up	Initial	5212C962-77E5-416B-8BE3-CE030286C451	Healthy	None	Enabled   Disable	Quarantine   Delete
172.31.70.85:8089	sh4	Up	Successful	5212C962-77E5-416B-8BE3-CE030286C451	Healthy	None	Enabled   Disable	Quarantine   Delete
172.31.72.220:8089	Deep-SHC	Up	Successful	5212C962-77E5-416B-8BE3-CE030286C451	Healthy	None	Enabled   Disable	Quarantine   Delete
172.31.74.232:8089	sh1	Up	Successful	5212C962-77E5-416B-8BE3-CE030286C451	Healthy	None	Enabled   Disable	Quarantine   Delete
172.31.76.49:8089	idx1	Up	Initial	5212C962-77E5-416B-8BE3-CE030286C451	Healthy	None	Enabled   Disable	Quarantine   Delete
172.31.76.309:8089	cmaster	Up	Successful	5212C962-77E5-416B-8BE3-CE030286C451	Healthy	None	Enabled   Disable	Quarantine   Delete

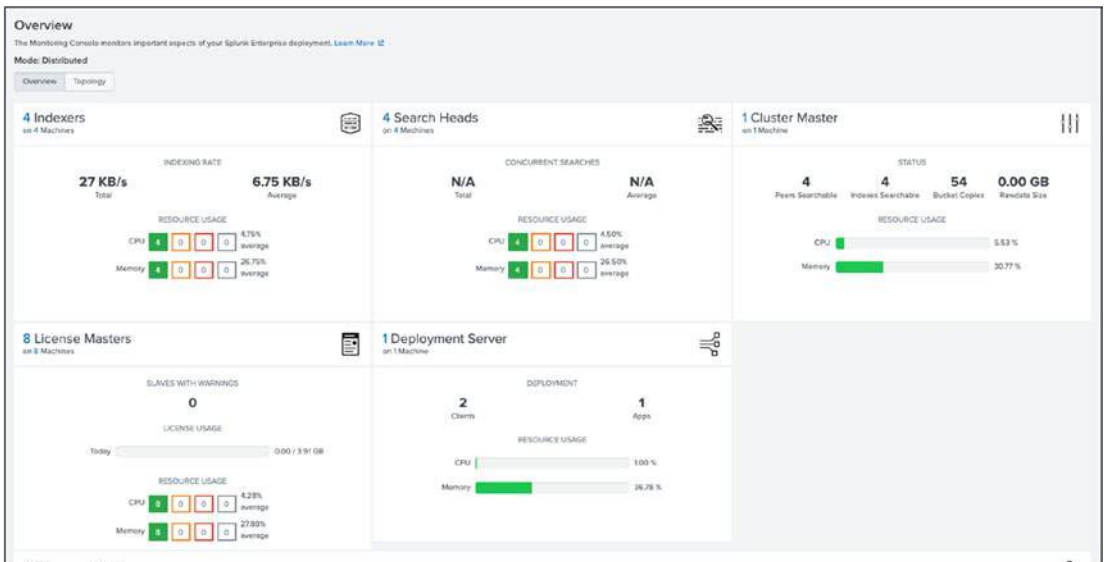
Figure 18-18. Monitor Console:Search Peer

## General Setup for Distributed Environments

To set up a general distributed environment in Splunk, go to the deployment server/ Monitoring Console instance in Splunk Web and refer to the following instructions.

1. Go to the deployment server/Monitoring Console and then to Settings in Splunk Web.
2. Go to Monitoring Console.
3. Click Settings and then go to General Setup on Monitoring Console.
4. Select Distributed under the Mode option. Click Continue. You should see the instances.
5. Examine the auto-selected roles. Edit the roles that are not configured according to their use case.
6. After editing all roles according to their instance, click Apply Changes.
7. Save all configured changes and go to the Overview page.

On the Overview page, you find the deployment diagram shown in Figure 18-19.



**Figure 18-19.** Monitor Console:Deployment Diagram

## Conclusion

In this chapter, you learned how to deploy the Splunk environment on the AWS platform.

This also brings you to the end of the book. I tried to cover all the Splunk topics, but if you want to learn even more, please refer to the Splunk docs at <https://docs.splunk.com/Documentation>. If you have concerns, or you are not able to solve an issue, go to the Splunk Community at <https://community.splunk.com/t5/Community/ct-p/en-us>. A Splunk expert can guide you and help you solve your questions.

# Index

## A

Ad hoc searches, 309

Admin exam set

authentication types, 271

bucket types, 268

clustering, 270

data integrity, 270

default configuration files, 267

deployment management, 267

forwarder type, 269

indexer operating system, 271

input phase, 269

local user accounts, 271

methods, 271

monitoring console monitor, 269

network inputs, 269, 271

props.conf file, 270

simultaneous searches, 268

single line event source, 267

universal forwarder, 268

whitelists/blacklists, 268

Agentless HTTP input

.conf file, 235, 236

event collector, 234, 235

parse data, 237, 238

receive events, 236, 237

web, 231–234

Alerts

cron expressions, 94–96

meaning, 92

steps, 92

test selection, 92

trigger, 93

Amazon Web Services (AWS)

deployment, 406

configuration, 413

forwarders, 413

internal logs, 416, 417

master node, 407

outputs.conf, 412

scalable platform, 409–411

server.conf, 411

universal forwarders, 411, 414–416

EC2 instances, 399

enterprise instance configuration,  
399, 400

forwarder, 400

infrastructure as a service (IaaS), 395

management console, 396

monitoring console, 417

distributed environment, 418

search peer, 417

multisite index clustering, 401

master configuration, 401

slave node, 402

overview, 395

platform as a service (PaaS), 396

search head

clustering, 404–406

configuration, 403, 404

software as a service (SaaS), 395, 396

## B

### Buckets

- bloom filters, 166
- data expiration, 164
- definitions, 164
- frozen bucket, 164, 165
- hot bucket, 163
- properties, 163
- search functions, 166
- search indexed data, 165
- .tsidx, 166
- warm/cold, 164

## C

### Capacity planning

- components, 276
- deployment
  - concurrent users, 277
  - dimensions, 276
  - incoming data, 276
  - indexed data, 277
  - searches, 277
- disk storage, 277

### Clustering, 185

### Cold buckets, 164

### Common Information Model

- (CIM), 101, 117
- definition, 117
- pivot selection, 118, 119
- root events, 117
- selection, 118
- steps, 119
- tes22 selection, 119, 120
- test data model, 120

### Configuration (.conf) file and diag

- anonymize data, 253–256

deploymentclient.conf file, 250

diag (*see* Diag creation)

fine-tuning inputs, 250–253

indexes.conf file, 246

inputs.conf file, 247, 248

meaning, 243

merging logic configuration, 256–259

nutshell, 243

outputs.conf file, 248, 249

props.conf file, 244, 245

transforms.conf file, 246, 247

troubleshoot, 260, 261

## D

### Dashboards

- categories, 137
- creation, 139
- dynamic form (*see* Dynamic form-based dashboards)
- panel screen, 138
- static real-time data, 133–137
- steps, 139
- types, 133

### Data models/pivot

- creation, 102–104
- datasets, 102
- predict sales patterns
  - auto-extraction states, 106, 107
  - child datasets inherit, 108
  - dataset process, 108
  - evaluate expressions, 105, 106
  - pie chart, 109–111
  - screenshot, 109
  - screenshot process, 110, 111
  - steps, 104, 105

### Debugging configuration files

- Btool, 260

- Deployment servers
    - app management issues, 347
      - irreversible, 347
      - lookup tables, 347
    - arbitrary content, 343
    - cluster master node, 349–352
    - configuration bundles, 352
    - directories, 344
    - forwarder management
      - forwarder management, 344–346
    - indexer discovery, 357–359
    - indexes, 349
    - infrastructure planning (*see*
      - Infrastructure planning)
    - load balancing, 355–357
    - master node, 350
    - meaning, 343
    - redeployment, 346
    - search head cluster, 352–355
    - serverclass.conf, 347
    - SOCKS5 proxy, 360, 361
  - Designing architecture
    - chart information, 292, 293
    - clustering type II M2/M14, 302
    - large-scale enterprise, 298–300
    - medium enterprise, 296–298
    - multisite indexer, 300–302
    - searching, 298
    - small-scale enterprise, 295
    - software components, 292
    - SVAs characteristics, 294
    - workflow, 294
  - Diag creation
    - command line, 261
    - creation screenshot, 262
    - node running, 261
    - tar.gz/diag.log file, 262
    - web/terminal/CLI, 261
  - Disk Size Calculation, 311–313
  - Dynamic form-based dashboards
    - drop-down menu, 145–147
    - link list, 147–150
    - radio button, 140–142
    - time modifier, 142–145
    - user interface (UI), 150, 151
- ## E
- Event action, *see* Workflow action
  - EXtensible markup language (XML), *see*
    - Dynamic form-based dashboards
- ## F
- Field aliases
    - data sources, 62
    - delimiters extract field, 64
    - field extraction, 65
    - images, 66
    - meaning, 62
    - new screen, 65
    - source type page, 63
    - steps, 62, 65, 66
    - testing page, 67
    - values, 66
    - working screen, 66
  - Field extraction
    - delimiters
      - renaming process, 58
      - steps, 57
      - structure data, 57
      - tab selection screen, 57
    - methods, 54
    - regular expressions
      - fields screen selection, 55
      - inline, 55, 56



## INDEX

### Field extraction (*cont.*)

- ports, 54
- steps, 54
- unstructured event data, 54

### Fine-tuning inputs

- props.conf, 252, 253
- source type web, 251, 252

### Forwarder management, 347–349

- apps/server classes/clients, 193
- client configuration, 195
- configuration, 193
- deployment servers, 344
- screenshot, 194
- serverclass.conf, 347
- tabs, 193

### Forwarding data, *see* Universal forwarder

## G

### GET workflow actions

- field value form, 113
- search option, 114
- spaces/special characters, 113
- steps, 112

## H

### Hardware/scaling considerations, 310, 311

### Health check monitors, 321–323

### Hot bucket, 164

### HTTP and Secure HTTP (HTTPS) protocols, 231

### HTTP Event Collector (HEC), 231

## I

### Index calculation, 307, 308

### Indexer clusters

### architecture, 196

### CLI, 200, 201

### .conf file, 199, 200

### configuration, 196, 197

### master/peer node, 196, 198, 199

### repetition factor, 195

### web creation, 197–199

### Indexer discovery, 357

### configuration, 357

### forwarders, 359

### master node, 358

### peer nodes

#### inputs.conf, 358

#### web, 358

### Indexes

### buckets, 172

### clustering

#### excess bucket copies, 375

#### maintenance mode, 373

#### offline command, 372

#### peer node/offline, 372

#### restart percentage, 374

#### rolling restart, 374

#### searchable option, 374

### command-line interface (CLI), 176, 177

### configuration file, 176

### data processing, 172

### data storage

#### CLI, 371

#### configuration file, 372

#### database, 369

#### disk usage, 371

#### size, 370, 371

### event configuration, 366

#### CLI, 367

#### configuration file, 366

### internal types, 173

- metrics indexes
    - CLI, 368
    - configuration file, 367
  - multisite index clustering (*see* Multisite index clustering)
  - parallelization
    - allocation method, 369
    - pipeline set, 368
    - server.conf file, 369
  - peer/master node, 375
  - removing/index data, 368
  - REST API endpoints, 378
    - configuration files, 380
    - functions, 378
    - job search, 379
    - methods, 378
    - searches, 379, 380
  - types, 173, 365
  - web creation, 173–176
  - Index parallelization, 311
  - Infrastructure planning
    - architecture (*see* Search head configuration)
    - capacity planning, 276, 277
    - meaning, 275
    - search peer, 278–280
      - CLI, 279
      - .conf file, 279
      - web, 278
  - Inline regular expressions, 56, 57
  - Input data
    - agentless input (*see* Agentless HTTP input)
    - forwarding options
      - code block command, 214
      - compresses raw data, 214
      - compressing data, 213
      - data input, 213
      - indexer acknowledgment, 213, 215
      - queue size, 214, 216
      - Securing the feed (SSL), 214–216
    - meaning, 213
    - monitor input, 217–221
    - scripted input, 221–225
- ## J
- journal.gz, 166
- ## K
- Key-Value Store (KV Store), 76
  - Knowledge managers/dashboards, 125
    - dashboards (*see* Dashboards)
    - objects global, 126–128
    - orphaned knowledge objects
      - monitor health checks, 130, 131
      - reassign objects page, 131
      - steps, 130
    - read/write permissions, 129, 130
    - responsibilities, 126
    - types, 125
    - visibility, 128, 129
- ## L
- Licenses
    - addition, 171
    - data types, 169
    - indexing phase, 169
    - management, 169
    - master/slaves, 169
    - pooling
      - creation, 172
      - meaning, 172

## INDEX

### Licenses (*cont.*)

- screen image, 168
- slaves, 170
- types, 167

### Light/heavy forwarders

- configuration, 191
- forward and receive data, 192
- instance process, 191
- types, 190

### Lightweight Directory Access Protocol (LDAP)

- authentication, 201
- roles, 205, 206
- strategies, 202–205

### Linux, 189, 190

### Load balancing

- forwarder forwards data, 355
- static configuration
  - CLI, 357
  - output.conf, 356
  - time, 356
  - volume, 356

### Lookups

- automatic lookup, 79–81
- creation, 77
- definitions, 78
- KV/CSV table, 76
- meaning, 75
- table file, 77, 78
- types, 76

## M

### macOS installation, 13–16

### Macros

- .conf file, 60–62
- creation, 58

search page, 61

web creation, 59–61

### Monitor input

- blacklist rules, 218
- definition, 217
- directories, 218
- files/directories, 217
- inputs.conf, 220, 221
- source type page, 219
- web process, 218–220
- whitelist rules, 218

### Multiple-choice questions (MCQ)

- access options, 158
- bucket file, 391
- building data models, 158
- CLI command, 390
- cluster component, 392
- commands, 156
- components, 155
- .conf file, 158
- data adding, 157
- 60-day trial version, 156
- delimiters, 157
- events, 159
- extraction configurations, 393
- field aliases, 158
- filtering commands, 156
- grouping command, 156
- indexer and forwarder, 390
- indexer cluster, 389
- indexes.conf, 392
- indexes data, 392
- internal index/logs, 392
- lookup types, 157
- macros, 157
- model acceleration, 158
- multiple search pipelines, 389

- parallelIngestionPipelines, 390
- planning, 393
- regular expression, 389
- search head cluster (SHC), 391
- search job inspector, 391
- server.conf file, 391
- server role, 390
- stand-alone machine, 155
- statistical trends, 156
- tags, 157
- user certification, 155
- Multisite index clustering
  - master node fails, 376
    - back up file, 376
    - peer and search head nodes, 377
    - standby server configuration, 376
  - master restarts, 377
  - peer node, 377

## N, O

- Network input
  - .conf files, 228
    - TCP network input, 228–230
    - UDP configuration, 230, 231
  - TCP/UDP port, 226
  - web/deployment, 226–228

## P, Q

- props.conf
  - .conf file, 244, 245
  - regular expression, 255, 256
  - SEDCMD command, 254
  - sed (stream editor) script, 254, 255
  - source type, 252, 253
  - syntax, 254

## R

- Redeployment
  - client, 346
  - content, 346
  - forwarder management, 346
  - serverclass.conf file, 346
- Reports
  - acceleration
    - creation, 87, 89
    - summary, 90
  - content selection, 86
  - meaning, 85
  - schedule page, 90, 91
  - searches, 87
  - visualization, 85

## S

- Scripted input
  - inputs.conf file, 224, 225
  - meaning, 221
  - network port, 226–231
  - web, 222–224
- Search head cluster
  - deployment servers, 352–355
- Search head configuration
  - architecture roles, 303
  - CLI, 283
  - clustering
    - architecture, 284
    - captain, 285
    - CLI, 286
    - component, 283–287
    - dynamic captain election, 285
    - roles, 285
    - search captain, 286
    - static search captain, 287

## INDEX

### Search head configuration (*cont.*)

- company XYZ, 303–305
- .conf file, 282, 283
- constituent structure, 280
- data input, 305–307
- designing architecture (*see* Designing architecture)
- disk size calculation, 311–313
- hardware/scaling considerations, 310, 311
- index calculation, 307
- multisite indexer clustering
  - benefits, 287–292
  - CLI commands, 290–292
  - .conf file, 288–290
  - searches, 287
- total disk size, 308
- user planner, 309, 310
- web, 281

### Search Processing Language (SPL)

- basics, 29
- Boolean operators, 30
- chart command, 38
- components, 29, 30
- eval command, 40
  - comparison/conditional functions, 40, 41
  - conversion function, 41
  - cryptographic functions, 42
  - date and time functions, 42, 43
  - informational functions, 43, 44
  - mathematical functions, 44
  - multivalued function, 44, 45
  - statistical functions, 45, 46
  - text functions, 46
  - trigonometry and hyperbolic values, 46, 47
- field command, 49

### filtering commands, 32

- dedup command, 33
- head, 33
- tail command, 34
- where command, 32, 33

### grouping results, 50

### lookup command

- function, 48
- input lookup, 48
- output, 48

### pipe operator, 28

### reporting commands, 34

- aggregate functions, 36
- chronological/timestamp order, 37
- event order function, 37
- history command, 35
- multivalued stats/chart functions, 37
- rare command, 34, 35
- stats command, 36
- table command, 35
- timechart functions, 37
- top command, 34

### rex command, 47

### sorting, 31, 32

### syntax coloring, 31

### timechart command, 39

### time modifiers, 29, 30

### transaction command, 50

### untable command, 38

### Search queries

- data visualization, 67
- meaning, 67
- output solution, 69, 70
- solution screen, 68
- timespan option, 70
- transaction events, 68

### Search workflow action

- field values, 115

- form information, 114
  - search command, 116
  - steps, 114
  - time chart, 116
  - Securing the feed (SSL), 214–216
  - Security Assertion Markup Language (SAML)
    - configuration, 206–208
    - meaning, 206
    - user roles, 209
  - serverclass.conf, 347
  - SOCKS5 proxy host, 360, 361
  - Software Development Toolkit (SDK)
    - operations, 381
    - Python, 381
      - command line, 382–384
      - splunklib.client file, 381, 382
  - Splunk
    - admin exam overview, 3
    - architecture, 11
      - components, 10
      - diagram, 9, 10
      - indexing data structures, 11
      - input data, 10
      - parsing pipeline, 11
      - search operations, 11
      - tasks, 12
      - Universal Forwarder (UF), 11
    - benefits, 9
    - blueprint, 6–8
    - data information
      - data adding, 21
      - folder reset, 20, 21
      - gear icon, 20
      - props.conf file, 21
      - set source type screen, 22
      - source type box, 23
      - test index, 24
      - test.txt file, 21
    - data types, 8
    - history, 8, 9
    - installation
      - attributes, 13
      - macOS, 13–16
      - Windows, 16–20
    - learning flow, 5
    - overview, 3
    - prerequisites, 5
    - props.conf file, 3
    - requirements, 4–6
    - structure, 4
  - Splunk Validated Architectures (SVAs), 294
  - Static real-time dashboards, 133
    - different categories, 137
    - Eastern US cities, 134, 136
    - HTTP method, 133, 136
    - HTTP status code, 136
    - Test2.csv, 133
    - web page, 134, 135
    - Western US cities, 134, 135
  - Supply Chain Management (SCM), 128
- ## T
- Tags
    - event type, 83, 84
    - meaning, 81
    - search bar, 82
  - Time-series index file (.tsidx), 166
  - Total disk size, 308, 309
  - Transmission Control Protocol (TCP), 226
  - Troubleshooting
    - clustering
      - buckets, 338

## INDEX

### Troubleshooting (*cont.*)

- master node, [337](#)
- multisearch affinity, [338](#)
- peer node, [337](#)
- search heads, [337](#)

### deployment

- forwarders, [336](#)
- indexers, [335–337](#)

### job inspector

- execution cost query, [331–334](#)
- execution cost/search job
  - properties, [331](#)

### license violation

- definition, [334](#)
- meaning, [334](#)
- steps, [334](#)

### log files, [324, 325](#)

### metrics.log file, [325–327](#)

- aggregator processor, [326](#)
- bucket\_metrics messages, [330](#)
- input pipeline, [326](#)
- messages, [328, 329](#)
- pipeline messages, [327](#)
- queue messages, [328](#)
- tcpout connection messages, [329](#)
- udpin\_connections messages,
  - [329, 330](#)
- winparsingqueue processor, [326](#)

### monitoring console

- assign/server roles, [320](#)
- forwarder connections, [323, 324](#)
- functions, [317](#)
- health check monitors, [321–323](#)
- multi-instance deployment, [318](#)
- overview page, [320](#)
- server roles, [319](#)
- single instance deployment, [318](#)

### overview, [317](#)

## U, V

### Universal forwarder

- command line, [187](#)
- forwarder management, [193–195](#)
- indexer, [186, 187](#)
- light/heavy forwarders, [190–192](#)
- Linux, [189, 190](#)
- meaning, [186](#)
- .msi file, [188](#)
- systematic procedure, [185](#)
- web interface, [186](#)
- Windows, [187](#)

### User Datagram Protocol (UDP) port, [226](#)

### User management

- inheritance/capabilities, [179](#)
  - capabilities screenshot, [180](#)
  - index access, [181](#)
  - inheritance, [179](#)
- native user, [177–179](#)
- roles/working structure, [177](#)

### User planner, [309](#)

## W

### Warm bucket, [164](#)

### Windows, [16–20](#)

### Workflow action

- GET, [112–114](#)
- meaning, [112](#)
- search action, [114–117](#)
- test.txt source, [112](#)
- types, [112](#)

## X, Y, Z

### XML

- time modifier, [142–145](#)