PRODUCTION MAIL SERVERS WITH POSTFIX AND DOVECOT

A COMPLETE GUIDE TO DESIGN, DEPLOYMENT, SECURITY, AND OPERATIONS

DESIGN
ARCHITECT ROBUST AND SCALABLE MAIL SYSTEMS

DEPLOY
STEP-BY-STEP INSTALLATION AND CONFIGURATION

SECURE
APPLY SECURITY BEST PRACTICES AND HARDENING

OPERATE
MONITOR, TUNE, AND MAINTAIN WITH CONFIDENCE

AUTOMATE
AUTOMATE TASKS AND SCALE WITH EASE

FROM FUNDAMENTALS TO HIGH AVAILABILITY
PRACTICAL EXAMPLES. REAL-WORLD SYSTEMS. PRODUCTION READY.

KEVIN LIU

Production Mail Servers with Postfix and Dovecot

A Complete Guide to Design, Deployment, Security, and Operations

Steve Publications

This book is available at
https://leanpub.com/productionmailserverswithpostfixanddovecot

This version was published on 2026-08-03

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2026 Steve Publications

Contents

A Complete Guide to Design, Deployment, Security, and Operations . . . **1**

Introduction **2**

Chapter 1: How Email Works **4**
- The Anatomy of an Email Message 4
- SMTP: How Mail Moves Between Servers 5
- IMAP and POP3: How Users Retrieve Mail 7
- DNS and Mail Routing: MX, A, AAAA, PTR Records 8
- The Complete Delivery Path 9
- Modern Email Security: SPF, DKIM, DMARC Overview 10

Chapter 2: Planning Your Deployment **12**
- Single Server Versus Split Architecture 12
- Operating System and Package Selection 13
- Choosing Your Authentication Backend 14
- Mailbox Storage Strategies 16
- Domain and Address Planning 17
- Capacity and Resource Requirements 18

Chapter 3: Installing and Verifying Postfix **21**
- Installing Postfix from Packages or Source 21
- Understanding main.cf Structure and Directives 21
- Basic Network Configuration: inet_interfaces and mynetworks 21
- Hostname and Domain Identity: myhostname and mydomain 21
- Local Delivery and Relay Restrictions 21
- Testing SMTP Connectivity and Sending Test Mail 22

Chapter 4: Configuring Dovecot for IMAP and POP3 . . . . . **23**
- Installing Dovecot and Understanding Its Architecture 23
- Protocol Configuration: IMAP Versus POP3 Versus LMTP 23

Mailbox Location and Namespace Setup 23
Authentication Mechanisms in Dovecot 23
User Home Directories and Permissions 23
Testing IMAP and POP3 Access 23

Chapter 5: Connecting Postfix and Dovecot with SASL 25
How SASL Authentication Works in Mail Systems 25
Configuring Dovecot as the SASL Backend for Postfix 25
The Submission Service: Port 587 Configuration 25
AUTH Mechanisms: PLAIN, LOGIN, CRAM-MD5 25
Securing Credentials and Preventing Relay Abuse 25
Testing Authenticated SMTP Submission 25

Chapter 6: Transport Layer Security with TLS 27
Why TLS Matters for Email Security 27
Obtaining and Managing Certificates 27
Configuring SMTP TLS in Postfix 27
Configuring IMAP and POP3 TLS in Dovecot 27
Cipher Suites, Protocols, and Modern Requirements 27
Certificate Renewal Automation with Let's Encrypt 27

Chapter 7: Virtual Domains and Virtual Users 29
Why Use Virtual Domains and Users 29
File-Based Virtual Domain Configuration 29
Database-Backed Virtual Users: MySQL and PostgreSQL 29
LDAP Integration for Enterprise Environments 29
Virtual Aliases and Catch-All Addresses 29
Mailbox Directory Structure and Permissions 29

Chapter 8: Mailbox Formats, Storage, and Quotas 31
Maildir Versus mbox: Trade-offs and Performance 31
Configuring Dovecot Mailbox Formats 31
Storage Backends: Local, NFS, Object Storage 31
Implementing Quotas with Dovecot 31
Sieve Scripting for Server-Side Filtering 31

Chapter 9: DNS Configuration for Deliverability 32
MX Records and Mail Routing Priority 32
Reverse DNS and PTR Record Requirements 32
SPF Records: Syntax, Mechanisms, and Best Practices 32

DKIM Setup with OpenDKIM or Postfix-DKIM 32
DMARC Policy Configuration and Reporting 32
Testing and Validating Your DNS Configuration 32

Chapter 10: Spam and Malware Filtering 34
Filtering Architecture Options: Inline Versus LMTP 34
Setting Up SpamAssassin with Postfix 34
Rspamd as a Modern Alternative 34
ClamAV for Virus Scanning Integration 34
Tuning Filters: Reducing False Positives and Negatives 34
Greylisting and Rate Limiting 34

Chapter 11: Advanced Delivery and Relay Configuration 36
Configuring Relay Hosts and Smart Relays 36
LMTP Delivery for Parallel Processing 36
Queue Management and Deferral Policies 36
Bounce Handling and Non-Delivery Reports 36
Content Restrictions and Size Limits 36
Transport Maps and Domain-Specific Routing 36

Chapter 12: High Availability, Clustering, and Scalability 38
Single Points of Failure and Risk Assessment 38
Active-Passive Failover with Heartbeat or Keepalived 38
Load Balancing IMAP and Submission Services 38
Shared Storage for Mailbox Redundancy 38
Scaling Postfix Across Multiple Instances 38
Geographic Distribution and Edge Relays 38

Chapter 13: Monitoring, Logging, and Performance Tuning 40
Understanding Postfix and Dovecot Logs 40
Structured Logging and Log Aggregation 40
Key Metrics: Queue Length, Latency, Error Rates 40
Monitoring Tools: Prometheus, Grafana, Nagios 40
Performance Tuning Postfix: Processes and Buffers 40
Optimizing Dovecot: Caching, Indexes, and Limits 40

Chapter 14: Troubleshooting Methodologies 42
A Systematic Troubleshooting Methodology 42
Reading and Interpreting Postfix Logs 42
Diagnosing Dovecot Connection and Authentication Issues 42

Testing SMTP Delivery: telnet, swaks, and mailq 42
Debugging DNS and Email Authentication Failures 42
Common Failure Patterns and Their Solutions 42

Chapter 15: Operations, Maintenance, and Automation 44
Backup Strategies for Mail and Configuration 44
Upgrading Postfix and Dovecot Safely 44
Security Hardening Checklist 44
Automation with Ansible, Puppet, or Chef 44
Containerized Deployment Considerations 44
Migration Strategies: Moving to a New Server 44

Conclusion: The Reliable Mail Server 46

References 47

A Complete Guide to Design, Deployment, Security, and Operations

This book teaches you how to design, deploy, secure, operate, and troubleshoot production mail servers using Postfix and Dovecot. It takes you from the fundamentals of email protocols through advanced topics like high availability, performance tuning, and automation. Every configuration is explained with its rationale and security implications, and every concept is demonstrated with complete, production-ready examples that work together as a cohesive system. Whether you are deploying your first mail server or managing a fleet at scale, this book serves as both a learning resource and an operational handbook.

Introduction

Email is one of the oldest protocols on the Internet and one of the most complex to run correctly. A message leaving your inbox may traverse half a dozen servers before arriving at its destination, each hop governed by standards written decades ago but still in active use today. Behind every delivered email sits infrastructure that must be reliable enough to handle critical communications, secure enough to resist abuse, and observable enough for you to diagnose problems when they occur.

Postfix and Dovecot are the two most widely deployed open source components for this job. Postfix serves as the mail transfer agent responsible for sending and receiving messages between servers using SMTP. Dovecot provides IMAP and POP3 access for end users and can also handle local delivery through LMTP. Together they form a complete mail system that powers everything from small business deployments to large hosting providers serving millions of users.

This book is organized to take you from understanding how email works to running a production-grade deployment. The first chapter explains the anatomy of an email message and traces its journey through DNS lookups, SMTP handshakes, authentication checks, and final delivery into a mailbox. Chapter 2 helps you make architectural decisions before installing anything: whether to run everything on one server or split roles across multiple machines, what authentication backend to choose, how to plan for growth.

Chapters 3 through 9 form the core of the book. They walk you through installing and configuring Postfix for SMTP, setting up Dovecot for IMAP and POP3 access, connecting them with SASL authentication, enabling TLS encryption, hosting virtual domains with database-backed users, choosing a mailbox storage format, and configuring all the DNS records required for reliable delivery including SPF, DKIM, and DMARC.

The remaining chapters cover advanced topics: spam and malware filtering integration in Chapter 10, relay configuration and queue management in Chapter 11, high availability and clustering in Chapter 12, monitoring and performance tuning in Chapter 13, systematic troubleshooting in Chapter 14, and operational procedures including backups, upgrades, and automation in Chapter 15.

A few principles guide the material throughout:

- Every configuration directive is explained with its purpose, not just shown as a snippet to copy. You will understand what each setting does so you can adapt it to your environment.
- Security is treated as a first-class concern. TLS encryption, authentication hardening, relay restrictions, and DNS-based email authentication are woven into the narrative rather than bolted on at the end.
- The configurations presented work together as a complete system. You will see how Postfix talks to Dovecot via Unix sockets, how filtering integrates with delivery, how databases map virtual users to mailboxes, and how each component depends on the others.
- Real-world operational concerns are addressed directly: logging, monitoring, backups, upgrades, migration strategies, and troubleshooting methodologies based on actual production experience.

The book assumes you are comfortable working on a Linux system: editing configuration files, running commands in a terminal, understanding basic networking concepts like IP addresses and DNS. You do not need prior experience with mail servers. Every mail-specific concept is explained from first principles before being applied.

By the end of this book you will be able to deploy a secure mail server from scratch, host multiple domains with virtual users, integrate spam filtering and virus scanning, configure proper email authentication records, monitor your system for problems, troubleshoot delivery failures systematically, and scale your deployment as requirements grow. You will also understand enough about how email works under the hood to diagnose issues that fall outside any cookbook solution.

Let us begin by understanding what actually happens when you send an email.

Chapter 1: How Email Works

Before configuring a single file or running a single command, you need to understand how email moves across the Internet. This chapter explains the protocols, components, and data structures that make email work. Understanding these fundamentals will make every configuration decision in later chapters clearer and help you diagnose problems when things go wrong.

The Anatomy of an Email Message

An email message has two distinct parts: the envelope and the content. Confusing these two is one of the most common sources of misunderstanding about how email works, so let us be precise.

The envelope contains routing information used by mail servers to deliver the message. It includes a MAIL FROM address (also called the return path or reverse path) that specifies where bounces and delivery status notifications should be sent, and one or more RCPT TO addresses that specify the intended recipients. The envelope is defined by RFC 5321 and is invisible to end users in their email clients.

The content is what the user sees: headers like From, To, Subject, Date, Cc, Bcc, followed by a blank line and then the message body which may include plain text, HTML, or attachments encoded in MIME format. The content headers are defined by RFC 5322. The From header displayed to the recipient does not have to match the MAIL FROM envelope address, and the To header does not have to match the RCPT TO addresses. This separation is intentional and enables features like mailing list distribution where a single message sent to one list address appears in individual subscribers' inboxes with different visible recipients.

Here is a simple example showing both parts:

Envelope:

- MAIL FROM: sender@example.com

- RCPT TO: recipient@gmail.com

Content headers:

- From: Alice Sender alice@example.com
- To: Bob Recipient bob@gmail.com
- Subject: Meeting tomorrow
- Date: Mon, 15 Jan 2024 14:30:00 +0000
- Message-ID: unique-id@example.com

The envelope sender and recipient are used exclusively for delivery routing. The content headers are for human readability and client display. Email authentication mechanisms like SPF check the envelope sender, while DKIM signs the content headers and body. DMARC ties both together by requiring alignment between the From header domain and the authenticated domain. Understanding this distinction matters because many configuration decisions depend on which part of the message you are working with.

SMTP: How Mail Moves Between Servers

SMTP stands for Simple Mail Transfer Protocol, defined in RFC 5321. Despite its name it is not simple, but it is remarkably effective and has been the backbone of email delivery since the early days of ARPANET.

SMTP operates as a conversation between a client (the sender) and a server (the receiver). The client initiates a TCP connection, the server responds with a greeting banner, and then they exchange commands and responses using human-readable text. Each command is followed by a numeric response code indicating success or failure.

A typical SMTP transaction looks like this:

```
Client connects to server on port 25
Server: 220 mail.example.com ESMTP Postfix
Client: EHLO mail.otherdomain.com
Server: 250-mail.example.com
Server: 250-PIPELINING
Server: 250-SIZE 52428800
Server: 250-STARTTLS
Server: 250 ENHANCEDSTATUSCODES
Client: STARTTLS
Server: 220 Ready to start TLS
(TLS handshake occurs here)
Client: EHLO mail.otherdomain.com
Server: 250-mail.example.com
Server: 250-PIPELINING
Server: 250-SIZE 52428800
Server: 250-8BITMIME
Server: 250 ENHANCEDSTATUSCODES
Client: MAIL FROM:<sender@otherdomain.com>
Server: 250 Ok
Client: RCPT TO:<recipient@example.com>
Server: 250 Ok
Client: DATA
Server: 354 End data with <CR><LF>.<CR><LF>
Client: From: sender@otherdomain.com
Client: To: recipient@example.com
Client: Subject: Test message
Client:
Client: This is the message body.
Client: .
Server: 250 Ok: queued as A1B2C3D4E5
Client: QUIT
Server: 220 Bye
```

Several important details are visible in this exchange. The EHLO command announces extended SMTP capabilities and triggers the server to list its supported extensions like STARTTLS for encryption, SIZE limits, and PIPELINING for efficiency. The MAIL FROM command sets the envelope sender, RCPT TO specifies each recipient (you can have multiple RCPT TO commands for a single message), and DATA begins transmission of the actual message content terminated by a line containing only a period.

SMTP operates on three standard ports with different purposes:

- Port 25 is used for server-to-server mail transfer. When one mail server delivers to another it uses port 25. This is the core SMTP port defined in RFC 5321.

- Port 587 is the submission port, defined in RFC 6409. Email clients (MUAs) use this port to submit outgoing mail to their configured mail server. Submission requires authentication and TLS encryption by convention.
- Port 465 was historically used for SMTPS (SMTP over implicit TLS). It was deprecated in RFC 8314 but remains widely supported for backward compatibility with clients that do not understand STARTTLS on port 587.

The distinction between ports 25 and 587 matters for configuration. Port 25 handles incoming mail from other servers and should never require authentication (remote servers cannot authenticate to your system). Port 587 handles outgoing submission from authenticated users and must enforce both TLS and authentication to prevent open relay abuse.

When a message is destined for another domain, the sending server performs an MX record lookup in DNS to find the recipient's mail servers, connects to the highest priority one, and delivers using SMTP on port 25. If delivery fails temporarily (server down, mailbox full) the message goes into a queue and Postfix retries according to configurable backoff intervals. If it fails permanently (user does not exist, domain has no MX records) a bounce message is generated for the sender.

IMAP and POP3: How Users Retrieve Mail

Once mail arrives at the destination server and is stored in a user's mailbox, the user needs a way to read it. Two protocols serve this purpose: IMAP (Internet Message Access Protocol, RFC 3501) and POP3 (Post Office Protocol version 3, RFC 1939).

IMAP treats the mailbox as a server-side resource that clients access over the network. Messages stay on the server by default, and clients synchronize their view of folders, flags (read/unread, starred), and message state across all devices. If you read an email on your phone using IMAP it shows as read when you check from your laptop because both are looking at the same server-side state. IMAP supports multiple folders, server-side searching, partial message fetching (headers only before downloading full content), and IDLE mode for push-style notifications when new mail arrives.

POP3 was designed in an era when users had one computer and wanted to download their mail locally. It downloads messages from the server to a

single client and typically deletes them from the server afterward (though most clients now offer a "leave on server" option). POP3 has no concept of folders or synchronization: each client sees only what it has downloaded. State changes made on one device are invisible to others because there is no shared server-side state.

For modern deployments IMAP is almost always the right choice. Users expect their mail to be available from multiple devices with consistent state, and IMAP provides that naturally. POP3 remains useful in specific scenarios: users who want a local offline archive as their primary copy, environments with very limited server storage where messages should be downloaded and deleted, or legacy clients that only support POP3.

Dovecot supports both protocols and can run them simultaneously on the same server. IMAP listens on port 143 (STARTTLS) and 993 (implicit TLS). POP3 listens on port 110 (STARTTLS) and 995 (implicit TLS). The encrypted ports are strongly preferred because they protect credentials and message content from network interception.

DNS and Mail Routing: MX, A, AAAA, PTR Records

DNS is the glue that holds email delivery together. Without proper DNS records your mail will not be delivered or will be treated as suspicious by receiving servers. Let us go through each record type.

MX (Mail Exchange) records tell the world which servers accept mail for a domain. When another server wants to deliver mail to user@example.com it looks up the MX records for example.com, sorts them by priority (lower number means higher priority), and attempts delivery starting with the highest priority server. A typical configuration:

```
example.com.      IN  MX  10  mail.example.com.
example.com.      IN  MX  20  backup-mail.example.com.
mail.example.com. IN  A   203.0.113.10
backup-mail.example.com. IN A  203.0.113.11
```

The primary server (priority 10) receives mail first. If it is unreachable the sending server tries the backup (priority 20). Multiple servers at the same priority are treated as equal and load is distributed among them.

If a domain has no MX records, senders fall back to looking up an A record for the domain name itself. This is not recommended practice because it blurs the line between web hosting and mail hosting, but it is valid per RFC 5321.

A records map hostnames to IPv4 addresses and AAAA records map to IPv6 addresses. Your mail server's hostname (the one listed in MX records) must have a working A record so that sending servers can connect to it. If you support IPv6 you should also provide an AAAA record.

PTR (Pointer) records implement reverse DNS. They map an IP address back to a hostname. When your mail server connects to another server on port 25 many receiving servers perform a reverse lookup on your IP address and verify that the resulting hostname matches what you announced in your EHLO command and that it has a forward A record pointing back to your IP. This is called forward-confirmed reverse DNS or FCrDNS. Missing or mismatched PTR records are one of the most common reasons for mail being rejected or marked as spam.

PTR records are configured by whoever controls the IP address block, typically your hosting provider or ISP. You cannot set them in your own DNS zone. Contact your provider to request a PTR record for your mail server's IP address that matches its hostname.

Additional DNS records important for email authentication will be covered in detail in Chapter 9: SPF (Sender Policy Framework) specifies which IPs are authorized to send mail for a domain, DKIM (DomainKeys Identified Mail) publishes public keys used to verify digital signatures on outgoing mail, and DMARC (Domain-based Message Authentication Reporting and Conformance) defines policy for handling messages that fail SPF or DKIM checks.

The Complete Delivery Path

Let us trace an email from sender to recipient inbox step by step. This end-to-end view ties together all the concepts introduced so far.

Alice types a message in her email client addressed to Bob at bob@gmail.com and clicks Send. Her client connects to her organization's mail server on port 587 using TLS encryption, authenticates with her credentials via SASL, and submits the message using SMTP submission (RFC 6409). The envelope sender is alice@company.com and the envelope recipient is bob@gmail.com.

The mail server accepts the message and examines the recipient domain gmail.com. It performs an MX record lookup in DNS and discovers that Gmail's mail servers are at alt1.gmail-smtp-in.l.google.com and several other addresses with various priorities. The server connects to one of them on port 25, negotiates TLS via STARTTLS, and delivers the message using standard SMTP.

Gmail's receiving server checks the message against its security policies: it verifies that the sending IP is authorized by company.com's SPF record, validates the DKIM signature if present, checks DMARC policy, scans for spam and malware using content filters, and examines the sender's reputation. If all checks pass Gmail accepts the message with a 250 response code and queues it for local delivery into Bob's mailbox.

Bob opens his email client which connects to Gmail's IMAP server on port 993 using TLS, authenticates, and sees Alice's message in his inbox because IMAP reads directly from the server-side mailbox. The entire journey may take seconds or minutes depending on network conditions and queue states, but each step follows well-defined protocols.

In this book you will be responsible for running a mail server that handles several of these steps: accepting submissions from authenticated users on port 587, receiving incoming mail from other servers on port 25, performing DNS lookups to route outgoing mail, storing messages in user mailboxes accessible via IMAP, and optionally filtering spam and malware before delivery. Understanding the full path helps you understand where your server fits in the ecosystem and what expectations other servers have of it.

Modern Email Security: SPF, DKIM, DMARC Overview

The email system was designed in a time when trust was implicit. Anyone could send mail claiming to be anyone else, and there was no built-in mechanism to verify identity. Over the decades three complementary standards have emerged to address this problem.

SPF (Sender Policy Framework, RFC 7208) allows a domain owner to publish a DNS TXT record listing the IP addresses authorized to send mail on behalf of that domain. When a receiving server gets a message claiming to be from @example.com it checks example.com's SPF record and compares the sending IP against the list. If the IP is not listed the SPF check fails, indicating possible

spoofing. SPF only validates the envelope sender (MAIL FROM), not the From header visible to users.

DKIM (DomainKeys Identified Mail, RFC 6376) uses public-key cryptography to sign outgoing messages. The sending server adds a cryptographic signature to specific headers and the body of each message it sends. The corresponding public key is published in DNS as a TXT record under a domain like selector._domainkey.example.com. Receiving servers fetch the public key and verify that the signature was created by someone who possessed the private key, proving both the sender's identity and that the message was not modified in transit. DKIM signs the From header domain among others, providing stronger authentication than SPF alone.

DMARC (Domain-based Message Authentication Reporting and Conformance, RFC 7489) builds on both SPF and DKIM by defining what receiving servers should do when checks fail. A DMARC record published at _dmarc.example.com specifies a policy: none (monitor only), quarantine (treat failures as suspicious), or reject (refuse delivery). It also requires alignment: for the message to pass, either SPF must pass with the envelope sender domain matching the From header domain, or DKIM must pass with a signing domain that matches the From header. DMARC additionally provides reporting mechanisms so domain owners can see who is sending mail using their domain and whether it passes authentication.

Together these three mechanisms form a defense-in-depth approach. SPF prevents unauthorized servers from sending as your domain. DKIM proves message integrity and authenticates the sender cryptographically. DMARC tells receivers how to handle failures and gives you visibility into abuse. Modern receiving systems like Gmail, Outlook, and Yahoo increasingly require all three for good deliverability, and domains without them face messages being silently delivered to spam folders or rejected outright.

Chapter 9 covers configuring SPF, DKIM, and DMARC in detail with specific DNS record syntax and testing procedures. For now the key takeaway is that email authentication is no longer optional if you want your mail delivered reliably, and it should be implemented as part of your initial deployment rather than added later as an afterthought.

Chapter 2: Planning Your Deployment

Before installing packages or editing configuration files, take time to plan your architecture. Good planning prevents painful rework later when you need to split a monolithic server into separate components, migrate storage backends, or redesign authentication mechanisms. This chapter walks through the major decisions you will face and explains the trade-offs so you can choose the right approach for your environment.

Single Server Versus Split Architecture

The most fundamental architectural decision is whether to run all mail services on one server or distribute them across multiple machines.

A single server architecture puts Postfix (SMTP), Dovecot (IMAP/POP3/LMTP), filtering software, and the authentication database all on one machine. This is simpler to deploy and manage: there are fewer network connections to troubleshoot, configuration changes affect everything in one place, and resource requirements are modest for small to medium deployments. For organizations hosting a handful of domains with hundreds or even low thousands of users, a single well-configured server handles the load without difficulty.

The drawbacks of consolidation become apparent as scale grows. All services compete for the same CPU, memory, disk I/O, and network bandwidth. A spike in IMAP connections from users checking mail can slow down SMTP delivery. A virus scanning operation consuming disk I/O can delay message acceptance. If the server goes down everything stops: no incoming mail, no outgoing mail, no user access.

A split architecture separates concerns across dedicated servers. Common patterns include:

- Separate MX servers running Postfix that accept incoming mail on port 25 and relay it to backend mailbox servers via LMTP. This isolates the attack surface of port 25 from user-facing services.

- Dedicated submission servers handling authenticated SMTP on port 587, keeping credentials and client connections away from the main MX infrastructure.
- Separate IMAP/POP3 servers running Dovecot that serve user mailboxes, allowing independent scaling based on connection patterns rather than message throughput.
- Dedicated filtering servers running SpamAssassin, Rspamd, or ClamAV that scan all traffic before it reaches mailbox storage.

Split architectures provide better fault isolation, more granular scaling, and stronger security boundaries. A compromised MX server cannot directly access user credentials if IMAP runs elsewhere. You can add IMAP capacity without touching SMTP infrastructure. But complexity grows: you must manage inter-server communication, synchronize configuration across machines, handle network partitions gracefully, and ensure shared storage or replication keeps mailboxes consistent.

For most deployments starting today the pragmatic choice is a single server with clear boundaries in mind. Run Postfix and Dovecot together on one machine but structure configuration so that splitting later would be straightforward: use LMTP for delivery instead of direct local transport (so mailbox servers can move independently), store user data in an external database rather than local files, and keep storage paths clean so they can be exported via NFS or moved to a different backend. This gives you simplicity now without locking you into a monolithic design forever.

Operating System and Package Selection

Postfix and Dovecot run on virtually any Linux distribution. Your choice of OS affects package versions, configuration file layouts, service management, and available extras like database connectors.

Debian and Ubuntu provide well-maintained packages with reasonable versions. Ubuntu's release cycle (LTS every two years) gives a predictable update schedule. Debian stable offers longer support windows but potentially older software. Both distributions split Postfix database support into separate packages: postfix-mysql, postfix-pgsql, postfix-ldap for database lookups, and dovecot-mysql, dovecot-pgsql, dovecot-ldap for Dovecot authentication backends.

Red Hat Enterprise Linux, AlmaLinux, Rocky Linux, and Fedora use RPM packaging. RHEL-family distributions include Postfix by default in minimal installations. Package versions tend to be conservative but stable. The configuration layout differs slightly from Debian: database support is often compiled in rather than separate packages, and service management uses systemd with different unit file conventions.

Arch Linux and derivatives track newer package versions more closely, which means access to the latest features but potentially less stability. If you want bleeding-edge Postfix or Dovecot features this may be attractive, but production mail servers generally benefit from tested, stable releases.

Key considerations when choosing:

- Version compatibility: Postfix 3.6+ introduced syntax changes for TLS protocol settings (using >=TLSv1.2 instead of !SSLv2,!SSLv3,...). Dovecot 2.4 is the current stable branch with significant improvements over 2.3. Check your distribution's package versions against the features you need.
- Package availability: Ensure database connectors (MySQL, PostgreSQL, LDAP) are available as packages rather than requiring compilation from source.
- Support lifecycle: Choose a distribution with a support window that matches your operational planning horizon. A mail server is infrastructure you do not want to migrate every eighteen months because the OS reached end of life.
- Familiarity: Pick an OS your team knows well. Mail server problems are hard enough without fighting unfamiliar package management or init systems.

This book uses Debian/Ubuntu-style paths and commands as the primary examples because they are widely deployed, but notes RHEL-family differences where they matter. The Postfix and Dovecot configuration concepts are identical across distributions regardless of packaging details.

Choosing Your Authentication Backend

Users must authenticate to send mail via submission (port 587) and to retrieve mail via IMAP or POP3. The question is where those credentials are stored and how the system verifies them. Four main options exist:

System users (/etc/passwd and PAM): The simplest approach for a single-domain server serving local Linux accounts. Each email user has a corresponding Unix account, authentication uses standard PAM modules, and mailboxes live in the user's home directory. This works fine if you have fewer than fifty users and do not need to manage mail accounts separately from system access. It becomes unwieldy at scale because every mail user needs a system account with a home directory, UID, and shell entry.

Passwd-file: Dovecot supports a simple flat file format resembling /etc/-passwd that stores usernames and hashed passwords without creating actual Unix accounts. This is useful for small virtual hosting setups where you want to avoid database complexity but still support multiple domains and users outside the system password file. Managing dozens or hundreds of users by editing text files becomes error-prone, so this approach has a practical upper limit around fifty to one hundred accounts.

SQL databases (MySQL/MariaDB or PostgreSQL): The most common choice for medium to large deployments. Usernames, passwords, domain membership, aliases, quotas, and other attributes are stored in database tables that both Postfix and Dovecot query. This enables web-based administration panels like PostfixAdmin or custom management interfaces. Changes propagate immediately without restarting services because queries happen at runtime. PostgreSQL offers stronger data integrity guarantees with foreign keys and transactions. MySQL/MariaDB is more widely used in existing hosting environments. Both are excellent choices; pick based on your team's familiarity and existing infrastructure.

LDAP (including Active Directory): Ideal for enterprise environments where user accounts already exist in a directory service. Integrating with LDAP or Active Directory means mail authentication uses the same credentials as system login, VPN access, and other services. Users do not need separate mail passwords. Group membership can control mailbox access and distribution lists. The complexity is higher because you must understand your directory schema, configure bind accounts securely, handle referrals and replication, and map LDAP attributes to Dovecot's expected fields. But if LDAP already exists in your environment it is almost always the right choice for authentication.

The decision tree is straightforward:

- If users already exist in Active Directory or LDAP, integrate with it.

- If you need web-based administration and plan to host more than a few dozen accounts, use MySQL or PostgreSQL.
- If you have fewer than fifty users and want maximum simplicity, try passwd-file.
- If mail users are identical to system users on a single domain, PAM authentication against /etc/passwd works fine.

You can also combine backends: authenticate some users against LDAP and others against a local database, or use PAM as a fallback if the primary backend is unavailable. Dovecot supports multiple passdb entries evaluated in order, giving flexibility for complex environments.

Mailbox Storage Strategies

Where mailboxes live on disk affects performance, reliability, backup strategy, and scalability. The physical storage path must be chosen carefully because migrating mailboxes after they grow large is painful.

Local filesystem: The default and simplest option. Mailboxes reside on the server's local disks under a dedicated directory like /var/mail/vmail or /home/vmail. This gives the best performance because there is no network latency for file access. It also simplifies permissions and avoids NFS-related locking issues. The limitation is that mailboxes are tied to a single physical machine: if that machine fails, accessing mail requires recovering from backup or having previously configured replication.

For local storage use a dedicated partition or volume for mailboxes rather than mixing them with the operating system and application files. This allows independent sizing, monitoring of disk usage, and separate backup policies. Consider using a filesystem optimized for many small files like ext4 or XFS, both of which handle Maildir's pattern of thousands of individual message files efficiently.

Network File System (NFS): Exporting mailbox storage via NFS enables multiple servers to access the same mailboxes from different nodes. This is useful for high availability setups where any IMAP server can serve any user's mail because all mailboxes are on shared storage. However, NFS introduces complications: locking semantics differ from local filesystems, performance depends heavily on network quality and NFS server capacity, and certain Dovecot features like mmap-based indexing may not work correctly over NFS.

If using NFS, configure it with appropriate options: use NFSv4 for better security and state management, tune attribute caching timeouts carefully (too aggressive causes stale data problems, too conservative kills performance), and prefer Maildir format over mbox or mdbox because Maildir is more tolerant of NFS race conditions. Test thoroughly under load before deploying to production.

Distributed storage solutions: For very large deployments specialized storage systems like GlusterFS, CephFS, or commercial SAN solutions provide scalability beyond what a single NFS server can handle. These add significant operational complexity and are typically only justified at scale (tens of thousands of mailboxes or more). The Dovecot mailing list contains extensive discussion of cluster filesystem experiences; in general simpler is better unless you have clear requirements that demand distributed storage.

SSD versus HDD: Mailbox storage is I/O intensive, particularly with IMAP where every folder open may read index files and message headers. SSDs provide dramatically better performance than spinning disks for the random access patterns typical of mail systems. If budget allows use SSDs for mailbox storage; it is one of the most impactful single upgrades you can make for user-perceived responsiveness.

Regardless of storage choice, ensure proper permissions: create a dedicated system user (commonly named vmail) that owns all virtual mailboxes, set its UID and GID consistently across all servers in your deployment, and configure both Postfix and Dovecot to use these same values so they can access the same mailbox files without permission errors.

Domain and Address Planning

How you organize domains, users, and aliases affects configuration complexity and operational overhead. Plan this structure before creating accounts.

Domain strategy: Decide which domains your server will host and how it relates to your existing DNS infrastructure. Each hosted domain needs MX records pointing to your mail server, SPF records authorizing your sending IPs, and eventually DKIM keys and DMARC policies. If you are taking over mail for domains previously hosted elsewhere, coordinate the DNS changes carefully: lower TTLs before the cutover, set up backup MX records if needed,

and monitor both old and new systems during transition to catch any messages routed to the wrong place.

For virtual hosting where you manage mail for multiple unrelated domains, ensure your configuration treats each domain independently for authentication (user@domain-a.com is distinct from user@domain-b.com even with the same local part) while sharing infrastructure resources like storage space and filtering rules.

Address conventions: Establish naming conventions for user addresses before creating them. Should addresses use first.last, first_initial.last, or username formats? How do you handle name collisions? What about department addresses like info@, abuse@, postmaster@? RFC 5321 requires that every domain provide postmaster@ as an address that accepts mail; many receiving servers test this during initial delivery to verify the domain is actively managed.

Aliases and forwarding: Plan how aliases will work. Common patterns include:

- Catch-all aliases that forward all mail for unknown addresses at a domain to a specific user (useful but dangerous because it also catches spam targeted at random addresses)
- Role-based aliases like abuse@, sales@, support@ that forward to one or more team members
- Personal aliases so users can receive mail at multiple addresses without maintaining separate accounts

Distribution lists: If you need mailing lists for groups within your organization, decide whether to implement them as simple alias expansions (every recipient gets a copy directly from your server) or with dedicated software like GNU Mailman that provides subscription management, archives, moderation, and per-list configuration. Aliases work fine for small static groups; Mailman or similar is better for larger dynamic lists where users subscribe and unsubscribe.

Address verification: Consider whether to reject mail for unknown recipients immediately (during the SMTP transaction) or accept it and generate bounces later. Immediate rejection saves resources by not queueing messages destined for non-existent users, but it requires Postfix to query your user database during every RCPT TO command, adding latency. Dovecot's LMTP integration with Postfix's reject_unverified_recipient feature provides a middle ground: Postfix caches verification results locally so subsequent checks are fast while still rejecting unknown recipients early.

Capacity and Resource Requirements

Right-sizing your server prevents both performance problems from under-provisioning and wasted expense from over-provisioning. Email workloads have unique characteristics that make simple CPU or memory estimates misleading.

CPU: Postfix is not particularly CPU-intensive for normal operations. SMTP transactions are mostly I/O waiting (network and disk). Dovecot's IMAP handling involves some CPU for parsing commands and searching indexes but is still dominated by I/O. The main CPU consumers are filtering software (SpamAssassin, ClamAV) and cryptographic operations (TLS handshakes). For a server hosting up to a few thousand users with moderate activity, 2-4 CPU cores is typically sufficient. Add more if you run intensive content filtering or have very high message volumes.

Memory: Postfix uses minimal memory per connection. Dovecot's memory usage scales with concurrent IMAP connections because each active session loads index data. A reasonable baseline is 2 GB for a small deployment, 4-8 GB for medium deployments with hundreds of concurrent users, more if you run filtering software on the same machine (ClamAV alone can use 1-2 GB). Ensure enough memory for disk caching: mail systems benefit enormously from having frequently accessed index files and message headers cached in RAM.

Disk space: Estimate based on user count times expected mailbox size. A typical business user might accumulate 1-5 GB over a year depending on attachment patterns and retention policy. For 100 users planning for 2 GB each means 200 GB of mailbox storage plus overhead for indexes (roughly 5-10 percent additional). Always provision more than your initial estimate because mailboxes grow unpredictably, and running out of disk space is one of the most disruptive failures a mail server can experience.

Disk I/O: This is often the limiting factor. IMAP operations generate many small random reads (opening folders, reading headers, searching). Mail delivery generates sequential writes. If you see high I/O wait times users experience slow folder opens and message listing. SSDs help dramatically here. Also consider separating mailbox storage from index files if your filesystem supports it: indexes are read frequently and benefit most from fast storage, while message bodies are large but accessed less frequently.

Network bandwidth: SMTP delivery is bursty rather than constant. Most servers spend time idle between messages. Ensure your network link can handle peak loads when many users check mail simultaneously in the morning or when bulk mailings go out. For typical deployments a 1 Gbps connection is more than adequate; even 100 Mbps handles hundreds of concurrent users without difficulty because email messages are small compared to video or file transfer traffic.

Connection limits: Plan for how many simultaneous connections your server must handle. Each IMAP client maintains a persistent connection using IDLE mode, so active users count against your connection limit continuously. Dovecot's default process limits may be too low for busy environments; you will tune these in Chapter 13. For now plan on at least one concurrent connection per active user plus headroom for burst activity.

The configurations throughout this book assume a medium-sized deployment: perhaps fifty to five hundred users across one to ten domains, running on a server with 4 CPU cores, 8 GB RAM, and 500 GB SSD storage for mailboxes. Adjust resource allocations proportionally for smaller or larger environments.

Chapter 3: Installing and Verifying Postfix

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Installing Postfix from Packages or Source

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Understanding main.cf Structure and Directives

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Basic Network Configuration: inet_interfaces and mynetworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Hostname and Domain Identity: myhostname and mydomain

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Local Delivery and Relay Restrictions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Testing SMTP Connectivity and Sending Test Mail

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 4: Configuring Dovecot for IMAP and POP3

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Installing Dovecot and Understanding Its Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Protocol Configuration: IMAP Versus POP3 Versus LMTP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Mailbox Location and Namespace Setup

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Authentication Mechanisms in Dovecot

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

User Home Directories and Permissions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Testing IMAP and POP3 Access

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 5: Connecting Postfix and Dovecot with SASL

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

How SASL Authentication Works in Mail Systems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Configuring Dovecot as the SASL Backend for Postfix

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

The Submission Service: Port 587 Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

AUTH Mechanisms: PLAIN, LOGIN, CRAM-MD5

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Securing Credentials and Preventing Relay Abuse

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Testing Authenticated SMTP Submission

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 6: Transport Layer Security with TLS

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Why TLS Matters for Email Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Obtaining and Managing Certificates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Configuring SMTP TLS in Postfix

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Configuring IMAP and POP3 TLS in Dovecot

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Cipher Suites, Protocols, and Modern Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Certificate Renewal Automation with Let's Encrypt

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 7: Virtual Domains and Virtual Users

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Why Use Virtual Domains and Users

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

File-Based Virtual Domain Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Database-Backed Virtual Users: MySQL and PostgreSQL

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

LDAP Integration for Enterprise Environments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Virtual Aliases and Catch-All Addresses

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Mailbox Directory Structure and Permissions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 8: Mailbox Formats, Storage, and Quotas

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Maildir Versus mbox: Trade-offs and Performance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Configuring Dovecot Mailbox Formats

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Storage Backends: Local, NFS, Object Storage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Implementing Quotas with Dovecot

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Sieve Scripting for Server-Side Filtering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 9: DNS Configuration for Deliverability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

MX Records and Mail Routing Priority

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Reverse DNS and PTR Record Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

SPF Records: Syntax, Mechanisms, and Best Practices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

DKIM Setup with OpenDKIM or Postfix-DKIM

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

DMARC Policy Configuration and Reporting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Testing and Validating Your DNS Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 10: Spam and Malware Filtering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Filtering Architecture Options: Inline Versus LMTP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Setting Up SpamAssassin with Postfix

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Rspamd as a Modern Alternative

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

ClamAV for Virus Scanning Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Tuning Filters: Reducing False Positives and Negatives

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Greylisting and Rate Limiting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 11: Advanced Delivery and Relay Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Configuring Relay Hosts and Smart Relays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

LMTP Delivery for Parallel Processing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Queue Management and Deferral Policies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Bounce Handling and Non-Delivery Reports

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Content Restrictions and Size Limits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Transport Maps and Domain-Specific Routing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 12: High Availability, Clustering, and Scalability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Single Points of Failure and Risk Assessment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Active-Passive Failover with Heartbeat or Keepalived

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Load Balancing IMAP and Submission Services

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Shared Storage for Mailbox Redundancy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Scaling Postfix Across Multiple Instances

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Geographic Distribution and Edge Relays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 13: Monitoring, Logging, and Performance Tuning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Understanding Postfix and Dovecot Logs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Structured Logging and Log Aggregation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Key Metrics: Queue Length, Latency, Error Rates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Monitoring Tools: Prometheus, Grafana, Nagios

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Performance Tuning Postfix: Processes and Buffers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Optimizing Dovecot: Caching, Indexes, and Limits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 14: Troubleshooting Methodologies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

A Systematic Troubleshooting Methodology

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Reading and Interpreting Postfix Logs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Diagnosing Dovecot Connection and Authentication Issues

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Testing SMTP Delivery: telnet, swaks, and mailq

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Debugging DNS and Email Authentication Failures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Common Failure Patterns and Their Solutions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Chapter 15: Operations, Maintenance, and Automation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Backup Strategies for Mail and Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Upgrading Postfix and Dovecot Safely

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Security Hardening Checklist

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Automation with Ansible, Puppet, or Chef

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Containerized Deployment Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Migration Strategies: Moving to a New Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

Conclusion: The Reliable Mail Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/productionmailserverswithpostfixanddovecot.