GETTING STARTED WITH HASHICORP

VAULT

Foreword by Armon Dadgar

ANUBHAV MISHRA

Getting Started with HashiCorp Vault

A hands-on guide on HashiCorp Vault for beginners.

Anubhav Mishra

This book is available at https://leanpub.com/getting-started-with-hashicorp-vault

This version was published on 2025-07-27

ISBN 978-1-7773924-0-6

Tweet This Book!

Please help Anubhav Mishra by spreading the word about this book on Twitter!

The suggested tweet for this book is:

I just bought "Getting Started with HashiCorp Vault" by @build1point0 on @leanpub https://thevaultbook.com/ #vault #book

The suggested hashtag for this book is #thevaultbook.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

#thevaultbook

This book is dedicated to my father Late Dr. Arun Kumar Mishra, and my mother Prof. Kusum Mishra.

Contents

Foreword

Being a modern developer is no easy task. The continuous focus on developer productivity has given rise to an explosion in the number of languages, frameworks, platforms, and tools at their disposal. This leads to a paradox where developers have so many choices and so many things to learn that it is difficult to begin.

The push towards DevOps and empowerment of developers to own application lifecycles means there is a universe of tooling that was previously built for operators that now they are expected to use as well. This empowerment is not without cost. If a developer has the ability to deploy to production on a whim, then they are responsible for the non-functional requirements like reliability and security as well.

Security generally is a topic that developers get coached into believing is fundamentally hard. Part of the challenge is a unique domain of jargon, especially with cryptography. This is an unfamiliar realm for most developers and made worse by the sensitive nature of trivial mistakes undermining the entire approach. Well known password managers make the news for broken cryptography, and as the saying goes, you only had one job.

This dynamic creates a tension between developer productivity and enablement, and the sensitive and critical nature of security. In our experience, the end result is that security gets ignored until there is a major breach because it is too daunting or challenging to tackle. This was a very personal issue for us at HashiCorp. When we created our first SaaS service, we quickly found ourselves with very sensitive customer data and sought out best practices to manage it securely.

We surveyed a range of companies, from small startups to large enterprises. The approaches varied from bad to worse. It was common to have credentials in plaintext everywhere, from source code to configuration management. Rotation of credentials was extremely rare, even in the cases of employees with privileged access leaving. Auditability of access was close to zero for most. This pattern was so common that we dubbed it “secret sprawl”, and we felt there had to be a better approach.

What started as an internal project to give our own developers a better solution, ultimately became the HashiCorp Vault product. We started with a two basic design goals for users:

- **Make it simple to use**. This was the foremost goal, because if it was challenging we knew developers would default to doing nothing or circumvent the system.
- **Right by default**. The second goal was to make sure the “right” thing was the default behavior, because we should not expect the users to be experts in security or cryptography.

These goals shaped our approach to a great extent. We wanted the API to be a simple REST/JSON over HTTP, so that it was easy to write clients and integration applications with it. We wanted a capable CLI that made it easy to interact with Vault without needing to deeply understand the API. From there, we built an intuitive UI that would help users get a better understanding of Vault and its full set of capabilities.

The initial starting point for Vault was as a basic key/value store, that allowed secrets to be stored but with the promise that everything was encrypted at rest and in transit. All access to those secrets would require clients to be authenticated, explicitly authorized, and with a full audit trail. The "right" behavior we enforced was a shift to explicit AuthN/AuthZ and an emphasis on least privilege. Today these would be considered pillars of a "zero trust" approach to security. Many users of Vault are probably unaware of zero trust and its implications, but are implicitly nudged to the same outcome.

Early on we felt that the architecture of Vault could solve more than just being a static key/value store. In the state of "secret sprawl" a very common challenge was the rotation of credentials and secrets. Most organizations have a very manual process of creating new credentials and orchestrating an update across multiple teams and systems. We felt Vault could solve this by decoupling the lifecycle of a credential from the policy that governs it. Instead of specifying an exact credential to use for a database, only specify the policy of grants that should be available and the time to live.

We introduced pluggable "secret engines" which enabled this dynamic behavior. These new secret engines allowed roles and policies to define how database credentials, SSH keys, TLS certificates, API tokens, or cloud credentials should be granted. Upon request, the policy is used to generate an on-demand credential that is unique to the client with a fixed lifespan. The credential can be revoked early manually and is automatically deleted at the end of the time to live.

These "dynamic secrets" made it simple to manage credential lifecycle. For users of Vault, we maintained simplicity by using the same API and approach for dynamic and static secrets. We provided additional metadata about the time to live and helper tooling to manage refreshing credentials. It has been a security best practice to rotate credentials often, but in practice most organizations fall short. Vault pushes this right behavior towards the default behavior by making it simple.

Support for secret engines was driven by dynamic secrets and simplifying the credential rotation workflow. However, we quickly found the flexibility of putting logic in the Vault request path allowed us to solve new use cases we had not anticipated. We had an internal use case to manage sensitive customer data, but potentially at a scale that was much too large to store directly within Vault. A common pattern to solve that is to decouple the encryption of the data from the storage of the data.
This pattern has a few challenges. If the cryptography is done by the client applications, then the client requires an encryption key and must implement the cryptography. Cryptography is notoriously subtle and simple mistakes could break the approach, so you don't want to

trust a broad swath of developers to get the implementation correct especially with your most sensitive data. Second, key management is even more difficult than cryptography! Most applications end up either hard coding their encryption keys or storing them in plaintext. Your encryption is ultimately only as secure as the key management.

Ideally client applications would use a high level API which allows them to specify the operation, such as encrypt or decrypt, and provide a reference to a key. The API would be responsible for implementing the cryptography and managing the underlying key material. This would reduce the risk of implementation errors, as the central implementation could be more easily vetted. It would also reduce the risk of a badly managed key getting exposed in plaintext. Critically, it would also allow us to rotate encryption keys without breaking the client.

This approach led us to build the “transit” secret engine in Vault. Transit engine allows an administrator to define a set of named keys, such as SSN or CreditCard, and to allow users to make API calls to perform logical operations with those keys. The user can request an encryption with the SSN key and provide the plaintext value. The transit engine uses the underlying SSN key to perform the encryption with an opinionated encryption algorithm that is not exposed to the user, and returns only the ciphertext. The client is free to store the ciphertext in a database or object store of their choice.

Thus the transit engine decouples the encryption of the data with the storage of the data. Applications are freed from implementing cryptography or managing any key material directly. They are also free to use any data storage system they want. The transit engine only interacts with the plaintext or ciphertext “in transit” through Vault, and thus the scalability challenge is limited to a much smaller number of underlying encryption keys. The API allows administrators to frequently rotate and decommission the keys without coordinating across multiple teams and systems, because the keying material is never exposed.

The transit engine is another example of looking at a common security workflow, storing sensitive data, and providing a solution that is both simple and right by default. Clients get a simple high level API to interact with that is familiar for most developers. Vault can be opinionated about the underlying algorithms and enables key management, so that we can follow the best practices.

We could spend more time talking about the various secret engines, but the point has been sufficiently made. Starting from the premise of simplicity and right by default, Vault has been designed to solve many different security challenges in an elegant way. As we talk about the future of the project, we continue to look for ways to push and make the system more capable and simpler.
Part of the challenge is an evolving landscape of technology and workflows. As users embrace more ephemeral platforms such as AWS Lambda and container platforms like Kubernetes and Nomad, Vault is building more integrations to make it seamless to operate in those environments.

At the same time, a large motivator for cloud adoption is the availability of managed services

which reduce the operational burden for end users. We've heard the demand from HashiCorp users as well and are heavily invested in the HashiCorp Cloud Platform (HCP). Through HCP, we will provide managed versions of our tooling across the public clouds. In many ways, HCP shares the same goal of making it simple for users to operate our software, while also deploying the best practices configuration by default. To us this is a logical extension of the original goals of Vault.

For readers of this book, I hope you find this to be a gentle introduction to HashiCorp Vault. While the product has a great breadth and depth of capabilities, I hope you find there is a consistent approach and that you can grow into Vault. As you learn more, I hope that it opens your eyes to new ways to solve security challenges, perhaps by decoupling concerns or by rethinking workflows.

– Armon Dagar, Co-founder and Co-CTO, HashiCorp

Preface

My first encounter with a computer was in kindergarten. I went to a school nestled in a small town in North India, where there happened to be only one computer for the whole school to use. I was in luck since it had Logo[1] installed on it. From the time I first saw the machine, I knew instantly that I was going to work with computers. After immense persuasion, my very generous mother got me my first computer in grade four - a custom assembled Intel Pentium 4 HT (Hyper-Threading) powered computer with Windows 98 on it. It wasn't long before I was taking the computer apart and putting it back together to learn about the hardware and software components.

After transferring to a boarding school, my interest in computers grew. Throughout school, I was actively involved in the computer club. I was part of the group that helped build the campus fiber installation, a rush I'll never forget. I learned about CISCO networking gear and how it was used to connect computers together.

I studied computer engineering at the University of Victoria[2] and was introduced to cloud computing during my internships. As a software engineer, I gravitated towards infrastructure and developer tooling. Even though I was interested in web software development I still had the interest to learn how the software ran on the servers. I then had the privilege to take an operations engineering role that helped me explore cloud infrastructure and all the challenges that come with it. I was part of a small team of operations engineers that were responsible for creating and managing the platform to run hundreds of services on thousands of servers on AWS at Hootsuite. As a small team, we started off with some Bash and Perl scripts but we had to learn how to automate provisioning and deployment of services at scale. It was around then that we were introduced to HashiCorp tools such as Vagrant, Consul, Packer, and Terraform.

Vagrant helped us create development environments and share them internally with other teams. Packer helped us build server images for AWS and VMs running internally on VMWare. Consul helped us with service discovery, storing configuration flags using the Consul KV store, and routing between services. Terraform helped us create and manage cloud environments. What we loved about the HashiCorp tools is that they followed the UNIX philosophy[3], where each tool was designed to solve a specific problem and they could inter-op with other tools that were being used at the company. I ended up creating a Terraform workflow tool called Atlantis[4] that allowed teams to collaborate on infrastructure using Git-based workflows.

[1] https://en.wikipedia.org/wiki/Logo_(programming_language)
[2] https://www.uvic.ca/
[3] https://en.wikipedia.org/wiki/Unix_philosophy
[4] https://runatlantis.io

At Hootsuite, we were constantly finding ways to do secrets and credentials management. We had static secrets that were managed manually by our teams. If they had to be rotated, we needed to do that manually too. At HashiConf 2015, HashiCorp announced Vault, a secret management solution. It was early days for Vault back then but the released feature set was useful to us at that time and we started experimenting with it. Five years later, Vault is a popular tool that is used by small startups all the way to the world's largest organizations.

My goal after joining HashiCorp as a Developer Advocate was to make HashiCorp tools accessible to everyone while advocating for our users internally. I worked on projects that enabled me to grow the ecosystem around HashiCorp tools. I helped start a project called secrets-store-csi-driver5 and wrote the initial version of the Vault provider. The provider allowed to fetch secrets from Vault and mount them into containers running inside a Kubernetes pod as a volume. Speaking at conferences like KubeCon + CloudNativeCon and running workshops at O'Reilly OSCON about Vault was also very insightful.

From my experience of advocating and working closely with Vault, I learned that Vault is a swiss army knife. It can be used in multiple ways to help secure applications and infrastructure. Since users have so many options with Vault, it can make it seem complicated and overwhelming in places. The goal of this book is to make Vault accessible to folks that are new to it. This beginner's guide to learn Vault will teach you how it can be used in various operational scenarios. For anyone who learns by doing much like me, this book features excellent hands-on examples to learn a range of concepts. I hope this book serves as a great starting point for your journey into HashiCorp Vault.

– Anubhav Mishra, 2020

Who is this book for?

This book is meant for developers, operations, and security engineers.

Versioning

Book version: v0.1.0
HashiCorp Vault version: v1.5.4

5https://github.com/kubernetes-sigs/secrets-store-csi-driver

About the Author

Anubhav Mishra is the Senior Director of Product Management at HashiCorp[6]. Previously, he worked in the Office of the CTO at HashiCorp as the Advisor to the CTO and Head of HashiCorp Labs (Product Incubation). He has a passion for emerging technology, developer advocacy, and synthesizing solutions that help people. He also worked at Hootsuite[7], where he created Atlantis[8] - An Open Source project that helps teams collaborate on Infrastructure using Terraform[9]. Anubhav loves working with distributed systems and exploring new technologies. He also loves open source software and is continuously finding ways to contribute to projects that excite him. That has led him to contribute to projects like Virtual Kubelet[10] (a CNCF project) and dapr[11]. He has worked on projects like the secrets-store-csi-driver[12] where he helped write the HashiCorp Vault provider[13] that allows users to fetch secrets from Vault and mount them as volumes in containers running in a Kubernetes pod. He often speaks[14] at conferences. In his free time, he DJs, makes music and plays football. He's a huge Manchester United[15] supporter.

6 https://hashicorp.com
7 https://hootsuite.com
8 https://www.runatlantis.io
9 https://terraform.io
10 http://virtual-kubelet.io
11 https://dapr.io/
12 https://github.com/kubernetes-sigs/secrets-store-csi-driver
13 https://github.com/hashicorp/secrets-store-csi-driver-provider-vault
14 https://anubhavmishra.me/speaking
15 https://www.manutd.com

Acknowledgements

I would like to thank my good friend Nicholas Jackson[16] for helping me write this book.

A huge thank you to my amazing partner, Rashmi[17]! Her unwavering support in every aspect of my life has been invaluable, and I especially appreciate her help design the book's wonderful cover.

Thank you to Pranay and Pranav Shrestha for encouraging me to write this book.

I would also like to thank my high school computer teachers: Mr. Chaunhan, Mr. Amit Mahajan, and Mr. Vijay Sissodiya for teaching me the wonderful ways of computer science.

[16] https://twitter.com/sheriffjackson
[17] https://rashmityagi.ca

Part 1 - Basics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

1. Introducing Vault

This chapter covers:

- Why application security and secrets management are essential
- Basic concepts behind Vault and the architectural principles behind a typical Vault cluster

In the last couple of years, there have been many high profile incidents where attackers have compromised a system which has led to a substantial loss of data. The widest-reaching of these was possibly the Equifax breach. In 2017, the computerized systems of Equifax were compromised leaking social security numbers and other data of over 143 million American citizens. Putting this into context, the population of America at the time of the breach was approximately 324 million, meaning this leak affected nearly half the population of the USA. Equifax was not the only high profile company. The UK airline, British Airways, was the victim of a sophisticated attack which resulted in the leak of 324,000 transactions containing personal and financial data. Most recently at the end of 2018, The Marriott Hotel group had its guest reservation database leaked which contained over 500 million records, for approximately 327 million guests. Contained in these records was a large amount of personally identifiable data. This data included name, mailing address, phone number, email address, passport number, and encrypted payment card numbers. The 8.6 million payment cards were encrypted using the AES128 standard; however, Marriott cannot rule out that both the data and the key used to encrypt this data were not compromised.

1.1 Application security facts and figures

It is without a doubt that the problems related to application security are increasing. Extracting some figures from Gemalto's Breach Level Index and IBM's Cost of a data breach both published in 2018 we can ascertain that:

- 214 records are compromised every second
- 2.2% of compromised records protected by encryption
- 65% of all cases are linked to identity theft
- $3.86 million is the average cost of a breach
- $350 million is the breach cost when over 50 million records are leaked

- 72% increase in attacks resulting in leaked data between 2017 and 2018

The majority of the cost which an organization will incur when suffering a breach are direct costs like hiring forensic experts, engaging a law firm, or offering victims identity protection. And indirect costs such as the allocation of resources, such as employees time to notify victims and investigate the breach, and the loss of customers.

In both of these cases following three basic principles can dramatically reduce the risk of data loss:

- Manage privileged access and other security practices, maintaining compliance
- Securely managing cryptographic keys and controlling data access
- Encrypt all sensitive data at rest and in motion

These factors are echoed in the Gemalto report, and they recommend that "the new perimeter is the data itself and the users accessing that data. Mindsets need to change to adapt to this reality, and IT professionals need to accept that breaches will occur and attach security to the data itself and the users."

Securing data is precisely where Vault can help, let's take a quick look at its major features and concepts.

1.2 What is Vault

Modern systems need access to lots of different types of secrets, database credentials, API keys for access to external systems. Then there are the operators of the systems; often they too need access to databases and API keys. The problem is managing who has access to what secret and being able to audit this access. Vault has been designed to play this role; it satisfies the requirement of providing secrets to modern systems and being able to control who has access to them.

Vault architecture

Vault runs in a typical client/server model; the server runs as a centralized application within your infrastructure. Regardless if you are using the Vault CLI, SDK, or another tool, you interact with the server is via the HTTP API. It can operate as a standalone server or in a clustered highly available (HA) mode, which is designed to minimize downtime and not for horizontal scalability. When running in HA mode, Vault servers have two states; they either operate as "standby" or "active." Only one server operates as the active node, the other servers in the cluster operate as hot standbys; in the instances where the active node fails then Vault automatically fails over to one of the standby servers, which then becomes the active instance.

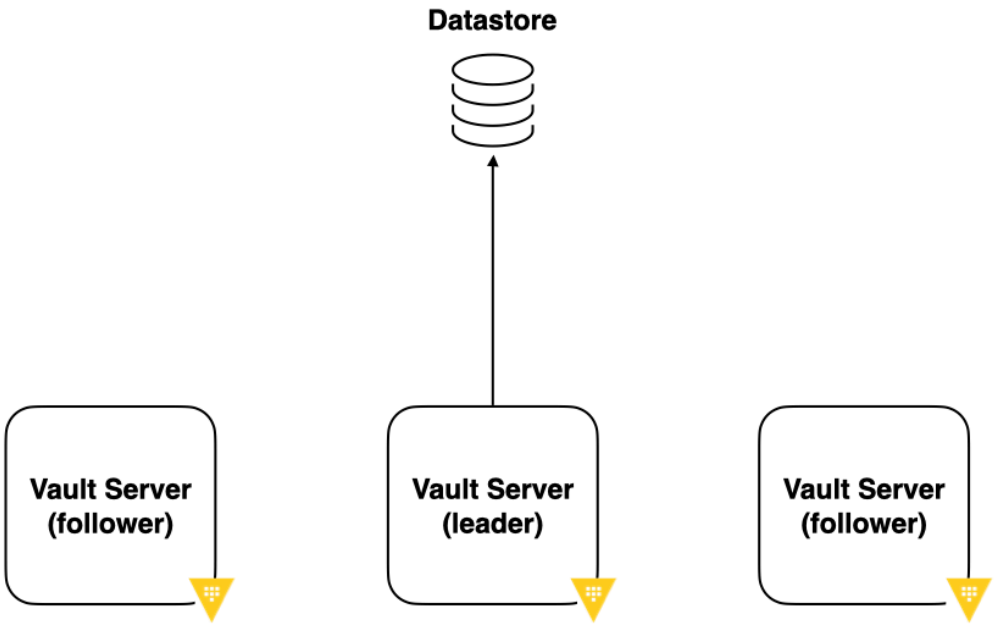

Figure 1.1 Vault's Highly-Available architecture diagram

The following are key features of Vault

Secure secret storage: Arbitrary key/value secrets are stored in Vault. Vault encrypts these secrets before writing them to persistent storage, so gaining access to the raw storage isn't enough to access your secrets. Vault can write to disk, Consul, and more.

Dynamic secrets: Vault can generate secrets on-demand for some systems, such as AWS or SQL databases. For example, when an application needs to access an S3 bucket, it asks Vault for credentials, and Vault generates an AWS keypair with correct permissions on demand. After creating these dynamic secrets, Vault will also automatically revoke them after the lease expires.

Data encryption: Vault can encrypt and decrypt data without storing it. Encryption in transit allows security teams to define encryption parameters and developers to store encrypted data in a location such as SQL without having to design their own encryption methods.

Leasing and Renewal: All secrets in Vault have a lease associated with them. At the end of the lease, Vault automatically revokes that secret. Clients can renew leases via built-in renew APIs.

Revocation: Vault has built-in support for secret revocation. Vault can revoke not only single secrets, but a tree of secrets, for example, all secrets read by a specific user, or all secrets of a particular type. Revocation assists in key rolling as well as locking down systems in the case of an intrusion.

While Vault is an incredibly capable piece of software with many features, the operational concepts are quite simple.

Workflow

The workflow for interacting with Vault builds around three core concepts, these are:

- Secrets
- Policy
- Authentication

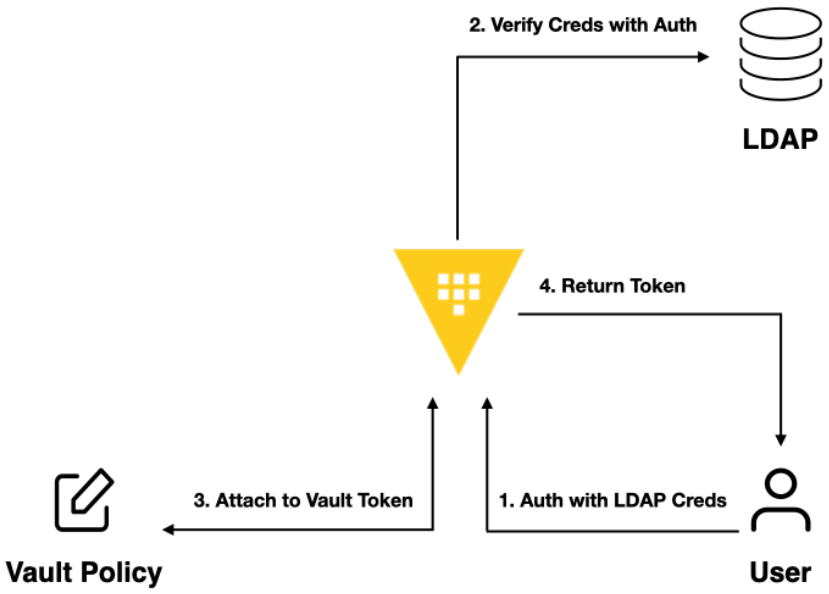

Figure 1.2 Vault interaction workflow

The following table presents these concepts in a little more depth.

Table 1.1 Vault concepts

Concept	Description
Secrets	Vault is capable of managing multiple different types of secrets; each secret type is managed by in Vault by an isolated component called a secret engine. Secrets engines are components which store, generate or encrypt data. Secrets engines are incredibly flexible, so it is easiest to think about them concerning their function. Secrets engines provided data, they take some action on that data, and they return a result. Some secrets engines store and read data – like encrypted Redis/Memcached. Other secrets engines connect to other services and generate dynamic credentials on-demand, provide encryption as a service, time-based one-time password (TOTP) generation, certificates, and much more.
Policy	The second core concept is Policy, every operation in Vault, be that a request to access secrets or to perform an administrative function is handled by at least one policy. Policies are “deny by default,” an empty policy grants no permission in the system, to gain access to secrets or to perform administrative functions a user or application must be assigned a valid policy. We mentioned earlier that all of the secrets and administrative operations in Vault are path-based. A path is one of the core concepts when writing policy. The path defines which feature the policy applies to; this can include glob-based Unix style pathname expressions for very coarse policy. It can also include explicit paths to create very granular policy.
Authentication	With Vault every action is secured with a Token. The tokens are assigned capabilities by Vault Policy which grants or denies access to the secret engines and various administrative functions in Vault. To obtain a token, the user must authenticate with Vault using the various Vault methods. Having multiple auth methods enables you

Concept	Description
	to use the auth method that makes sense for your use case and your organization. For example, an application running in Kubernetes might use the Kubernetes auth which uses the Pod's Service Account to authenticate with Vault. This method would not make sense for an operator so they could use GitHub or LDAP authentication.

Policy

These concepts come together in a typical workflow. Vault is just like any other web application. Just like in a web application, a user needs to authenticate to log into the application. A Vault user or application is also required to authenticate with Vault and get assigned policy in order to store/retrieve secrets or configure Vault. In the first step, the user authenticates with Vault using their LDAP credentials. Using the configured LDAP authentication method, Vault verifies these credentials with the configured backend. If the credentials are valid, Vault creates a token and attaches the policy relating to the user or group to it. It then returns this token to the user which is used for subsequent requests.

When a user requests a secret they pass this token as part of the request; Vault validates that the requested action is allowed by checking the policy attached to the token. Only if the policy is valid will the operation be allowed. Regardless of the user being a human operator or an application, this workflow remains the same.

1.3 Case studies

Let's take a more in-depth look at two typical application security problems and how Vault can help using the following two case studies.

Case study one: application security

Marionette is a significant global hotel chain; to reduce costs and satisfy the growing demand for online services it launched a new online reservation system. The system predominately wrapped an existing legacy system which was protected by physical hardware security with an online interface allowing self-service access to booking and payment systems. To keep this credit card and personally identifiable data safe they decided to build their own encryption service which would partially encrypt and decrypt data such as credit card information. In addition to this, they would invest further in physical security such as firewalls, web application firewalls, and intrusion detection systems.

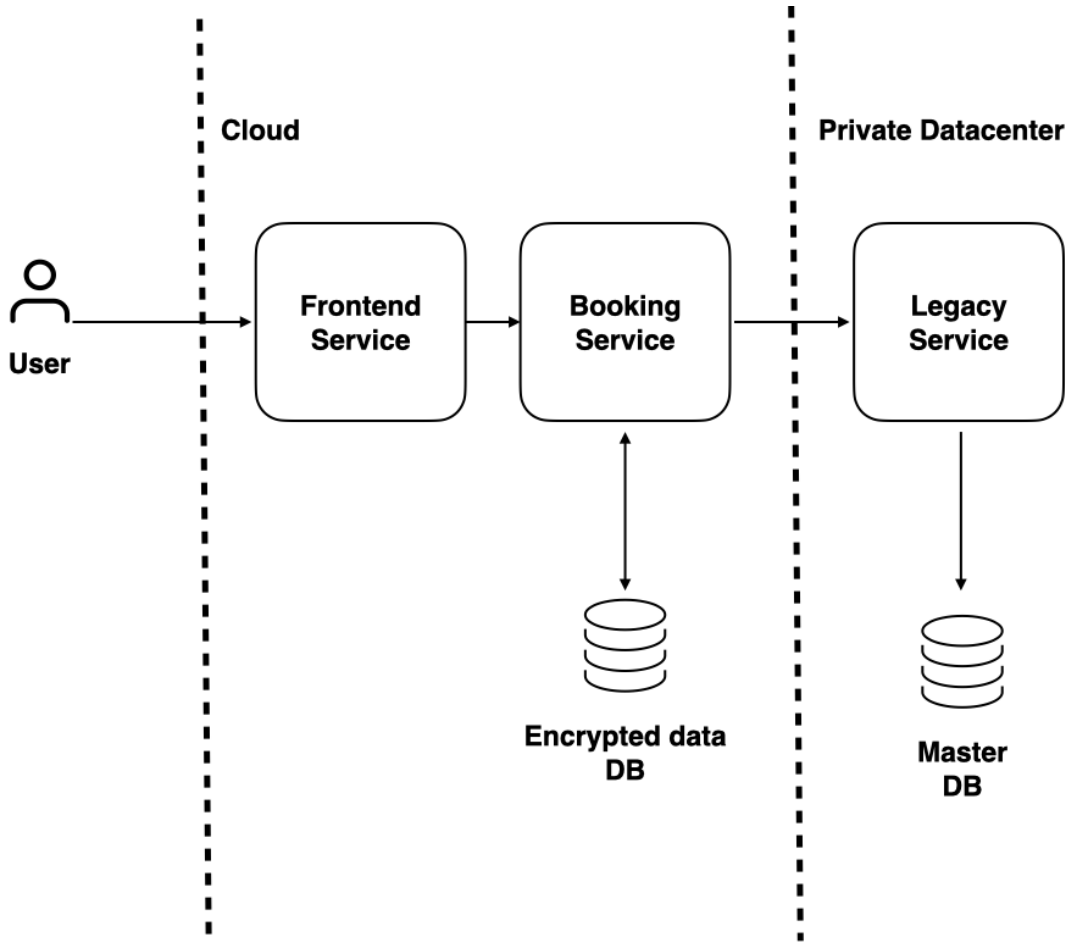

Figure 1.3 Architecture version 1

Everything went well the new system which was built using modern microservice principles was brought online, the customers and staff were delighted with the new modern system. Unfortunately in late 2018 intrusion detection systems notified the operations team that there had been unauthorized access in the booking and payments system. After a lengthy investigation, it was determined that again an application code level vulnerability in the front end favorites service had allowed an attacker to compromise the perimeter defenses and execute further attacks on the system. The attacker managed to gain access to the backend booking system and had successfully retrieved both encrypted and non-encrypted data which included guest names and addresses, passport information and some credit card information.

Some of this information had been encrypted however due to the way that the encryption was implemented, the key required to encrypt and decrypt the data was stored with the application which was using it. Marionette are currently unsure if the encryption keys have also been retrieved in the attack however the team who have been investigating the incident believe this was highly likely.

A report was prepared for the CISO which summarized the findings which led to this attack, these were:

- Insecure implementation of encryption for sensitive data
- No key rotation for this data, the original key had been used for over 4 years
- Widespread access to the encryption key, many applications, developers and operators had access to decrypt and read data in the production database
- Lack of encryption in transit, all internal traffic was transmitted and received in plain text. While all data transmitted over the public internet was correctly secured using TLS, it was assumed that a perimeter firewall-protected internal data

Based on the findings it was apparent that immediate remedial action was required to prevent a further attack, it also highlighted the fact that while the application development and operations teams were world-class, they were not cryptography experts. The external forensic experts who were assisting with the investigation and preparation of the report advised the team to implement Vault as a central secrets management platform. Vault would be used in the following areas:

- Encryption as a service, applications would no longer perform encryption functions, all requests to encrypt/decrypt data would be handled centrally by Vault
- Encryption keys would be stored and managed by Vault, access to these keys would be controlled and audited through granular access policy
- Transport level encryption would be rolled out through all internal services; Vault would be used to generate and manage X509 certificates required to secure application endpoints

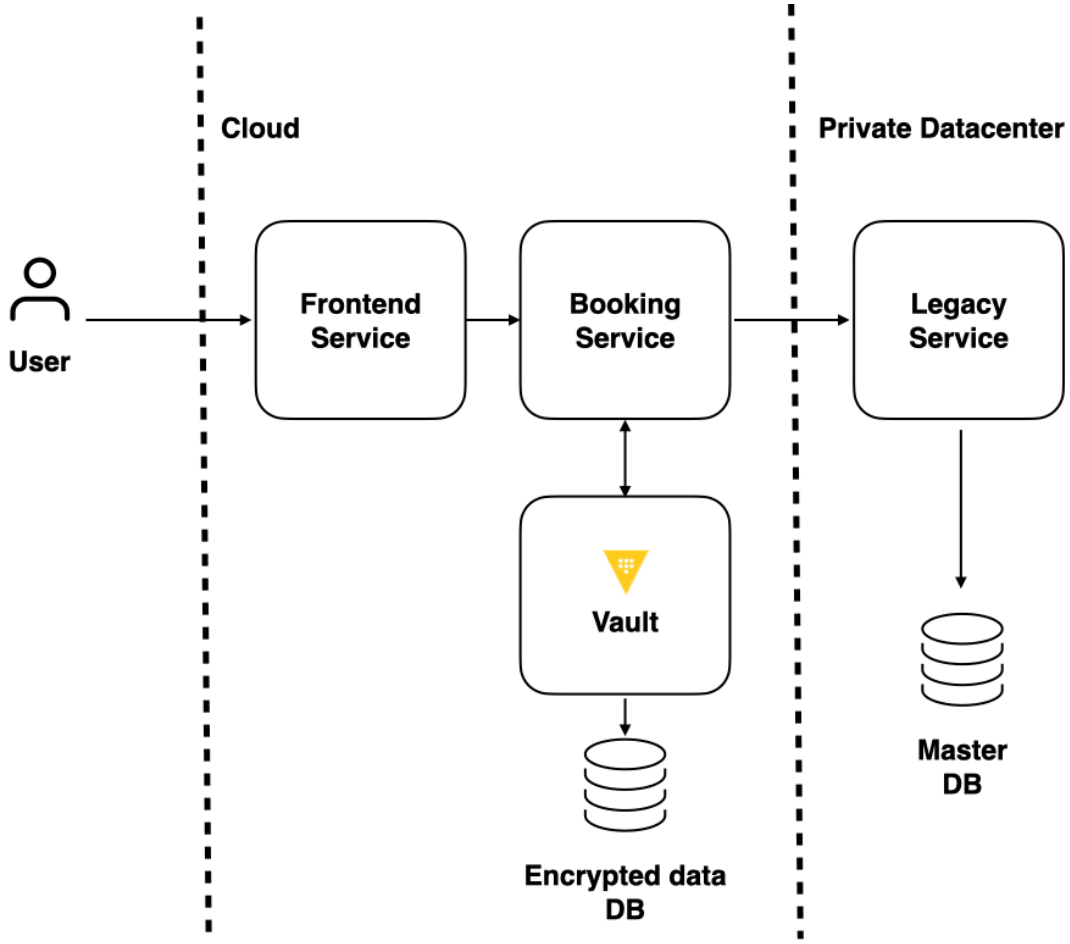

Figure 1.4 Architecture version 2

The learning curve for Vault was short since it is built around 3 simple principles of Authentication, Policy, and Secrets. Both the operations team who rolled out the cluster and the application development team embraced the change. The external auditors and the CISO were satisfied that the application had been secured and in the event of another breach their customer's data would be secure.

Case study two: operational security

Rohit works for Zenith, a massive IT consulting firm. As an operations engineer, Rohit required the highest level of access to live systems to provision and debug systems that are failing, or to perform maintenance tasks like updating software. He has access to SSH keys that enable him to login into all the application servers, and all of the databases servers in the production environment. In addition to this, all of the databases are configured using a global set of credentials which is known by Rohit and other members of the team.

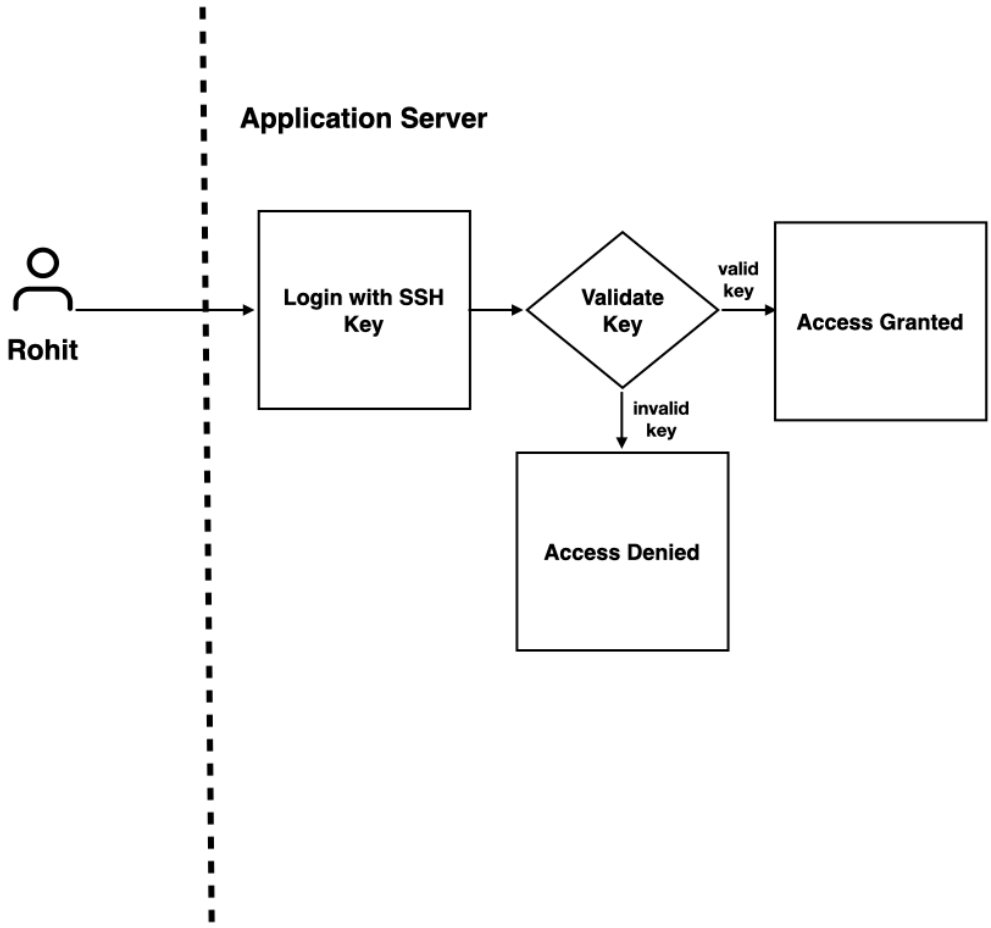

Figure 1.5 Authentication with SSH private key

After ten years of working at Zenith, Rohit decided to move on to his next endeavor; this meant that all his credentials need to be removed from all of the application and database servers. For an IT organization with thousands of applications and tens of thousands of servers, this is a difficult task to undertake. Given the difficulty of the job in hand, the operations engineering team concluded that it would take them almost five months to remove Rohit's permissions and to rotate all of the database credentials. During those five months, Rohit had access to live production systems that run customer applications for Zenith and access to all of their data.

The access Rohit has poses a severe security risk to both Zenith and their customers, users that leave an organization shouldn't have access to production systems, and revoking and removing this access should not be such a complicated task. Based on the conclusion of the operations engineering team it was evident that it would take a long time to rotate credentials since they did not have a centralized secrets management solution. A group of external security experts who were given the task to remediate and help with this process advised the team to implement Vault as it would enable Zenith to:

- Dynamically generate database credentials, SSH access, and access to cloud providers. Applications and operators would leverage Vault to dynamically generate time-bound credentials for accessing databases and for accessing production servers.
- Leverage their existing LDAP solution to authenticate users with Vault; this enables them to map LDAP groups to Vault policies without needing to create new methods to track organizational level access.
- Revoke access for users when they leave the organization, any credentials and access privileges generated through Vault can be automatically and immediately revoked.

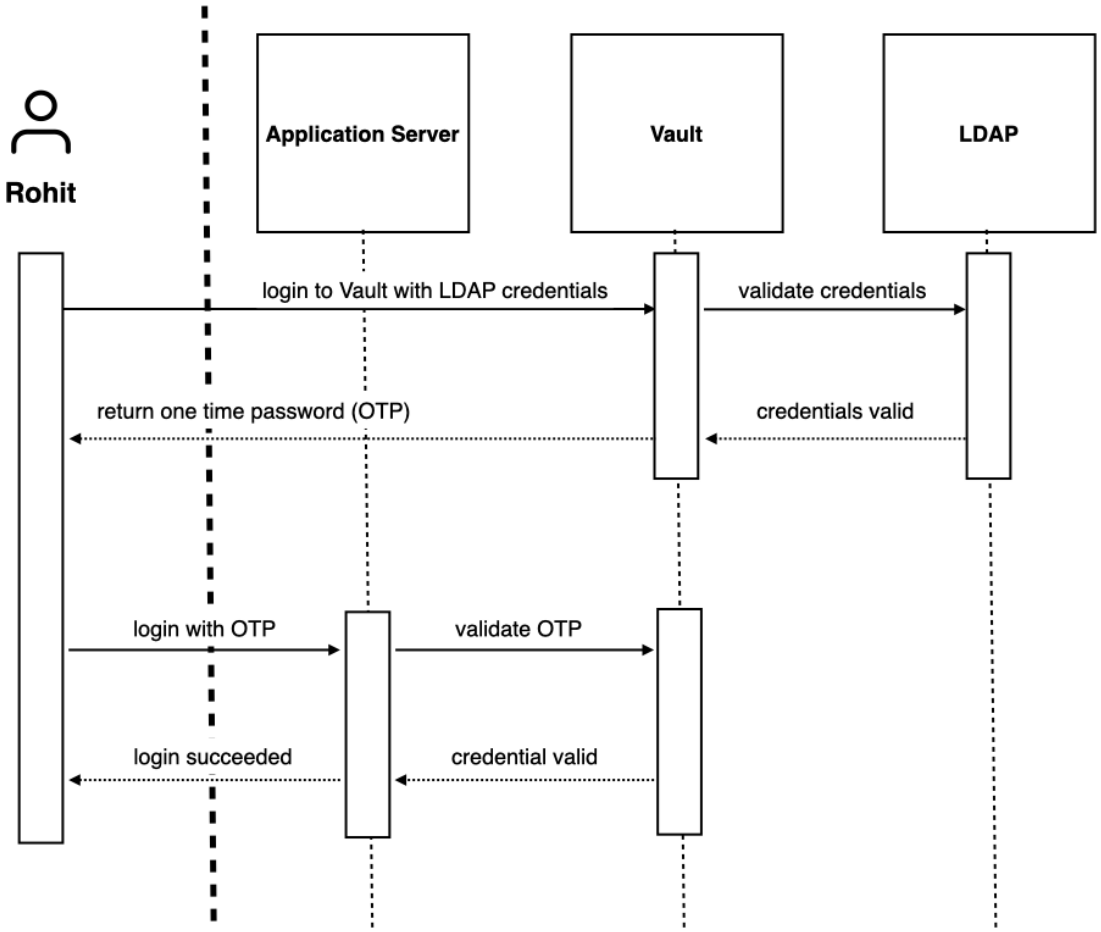

Figure 1.6 Authentication with Vault

Vault was easy to implement at the Zenith organization. Both operators and developers saw the benefits of using Vault as the central secret management platform. The external security team and the auditors were pleased with the implementation of Vault as it immediately gave them the capability to revoke access to production systems when an employee leaves the organization.

1.4 What you'll learn

In this book, we will cover the basic Vault concepts that are essential features of Vault such as Static and Dynamic Secrets, Authentication, and Encryption in Transit. You will learn how to configure, deploy, and operate it. We will also present application and operational patterns for leveraging Vault's powerful features from your applications. By the end of this book you will have learned how to:

- Use Vault as a centralized secrets management system to do the following:
  - Provide secrets to modern "cloud-native" and traditional applications
  - Store sensitive data like credit card and other personally identifiable information
  - Provide dynamic and time-limited credentials for data stores, cloud accounts, and SSH authentication.
- Implement correct and secure Identity and Access Management in Vault, providing granular access to secrets and administrative functions.
- Use Vault to provide cryptography as a service for applications using the transit backend.

In chapter 2 you'll look at Vault basics, including getting things set up in a development environment.

1.5 Summary

- Vault provides a mechanism to centralize secret management and gives modern applications access to secrets.
- Vault architecture consists of a typical server/client model.
- The workflow associated with interacting with Vault consists of secrets, policy, and authentication concepts.
- Vault can help with application and operational security in an organization.

2. Getting started with Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

2.1 Vault in practice

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Vault command-line interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Writing secrets into Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Reading secrets from Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Restricting secret access using policy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Vault HTTP API

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Accessing secrets using the Vault API

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Vault UI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

2.2 Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

3. Authenticating with Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

3.1 Basic authentication workflow with Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

3.2 Auth methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

User permissions in an organization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Enabling an authentication method

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring an authentication method

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Testing your LDAP server settings

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Mapping LDAP groups to Vault policy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Authenticating applications with Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring AppRole authentication method

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Authentication using the Vault Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Interacting with Vault using the Vault Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

3.3 Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

4. Storing and generating secrets using Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

4.1 Secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Enabling a key-value secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Generic key-value secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Storing a key with version 1

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Retrieving a key with version 1

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Versioned key-value secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Enabling a versioned key-value secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Storing versioned API keys

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Retrieving versioned API keys

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Database secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Enabling a database secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring the MySQL database plugin

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring a role for database secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Generating a dynamic database credential

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Revoking database credentials

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Generating on-demand cloud credentials

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

AWS secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Google Cloud secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

4.2 Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

5. Control access in Vault using policy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

5.1 Vault policy syntax

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Managing policies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

5.2 Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Part 2 - Application Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

6. Application secrets with Vault Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

6.1 Vault Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring Vault Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Creating templates for Vault Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring Vault Agent to render templates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

6.2 Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

7. Using Vault with HashiCorp Nomad

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

7.1 Using Nomad and Vault together

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Deploying MySQL

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring database secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring Vault with Nomad cluster-related policies and roles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring Nomad to use Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

7.2 Running workloads on Nomad

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Redeploying MySQL

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Deploying an application

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Validating application

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Revoking generated database credentials

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

7.3 Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

8. Using Vault with Kubernetes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

8.1 Using Kubernetes and Vault together

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring key-value secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Creating Vault service in Kubernetes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Testing Vault Connectivity from Kubernetes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Vault Agent Injector in Kubernetes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring Kubernetes authentication method

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring Vault policy and role

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

8.2 Fetching secrets for Kubernetes workloads from Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

8.3 Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Part 3 - Operational Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

9. Securing SSH with Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

9.1 Enabling SSH secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

9.2 Configuring SSH secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Creating a role to use the SSH secrets engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring machines to use one-time SSH passwords from Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing and configuring `vault-ssh-helper` program

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Configuring `sshd` and PAM configurations to use Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Using Vault's SSH secrets engine to login to remote machine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Revoking lease in Vault to disable SSH logins

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

9.3 Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

10. Integration Vault with CI systems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

10.1 Authenticating CI/CD system with Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Enabling and Configuring AppRole authentication method

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Store GitLab Token in Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Create Policy for Accessing Secrets

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Generating Role ID and Secret ID

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

10.2 Configuring Vault Plugin in Jenkins

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

10.3 Fetching Secrets in Jenkins Job

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Create Jenkins Job

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Run Jenkins Job

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

10.4 Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Appendix A - Vault development environment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Vault

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Starting the Vault Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Appendix B - GLAuth LDAP server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing GLAuth

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Downloading GLAuth config

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Appendix C - jq Command-line JSON Processor

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing jq

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Try it out

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Appendix d - Vagrant environment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Vagrant

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Vagrant provider

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Validate Vagrant installation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Install vbguest plugin

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Initialize Vagrant

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Start Vagrant machine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Access the Vagrant machine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Appendix e - Nomad development environment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Prerequisites

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Vagrant

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Docker in Vagrant

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Nomad

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Validate Nomad installation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Initialize Nomad Cluster in Dev Mode

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Validate the Nomad Dev Cluster

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Accessing Nomad UI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Appendix f - minikube environment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Prerequisites

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Docker

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Validating install

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing `kubectl` and Helm CLI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Validating install

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Install HashiCorp Helm repository

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing minikube

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Starting a Kubernetes cluster

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Validating Cluster

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Validating `kubectl`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Appendix g - Jenkins on Docker

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Prerequisites

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Docker

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Validate install

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Running Jenkins container

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Create local Jenkins directory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Starting Jenkins container

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Accessing Jenkins locally

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.

Installing Vault plugin

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/getting-started-with-hashicorp-vault.