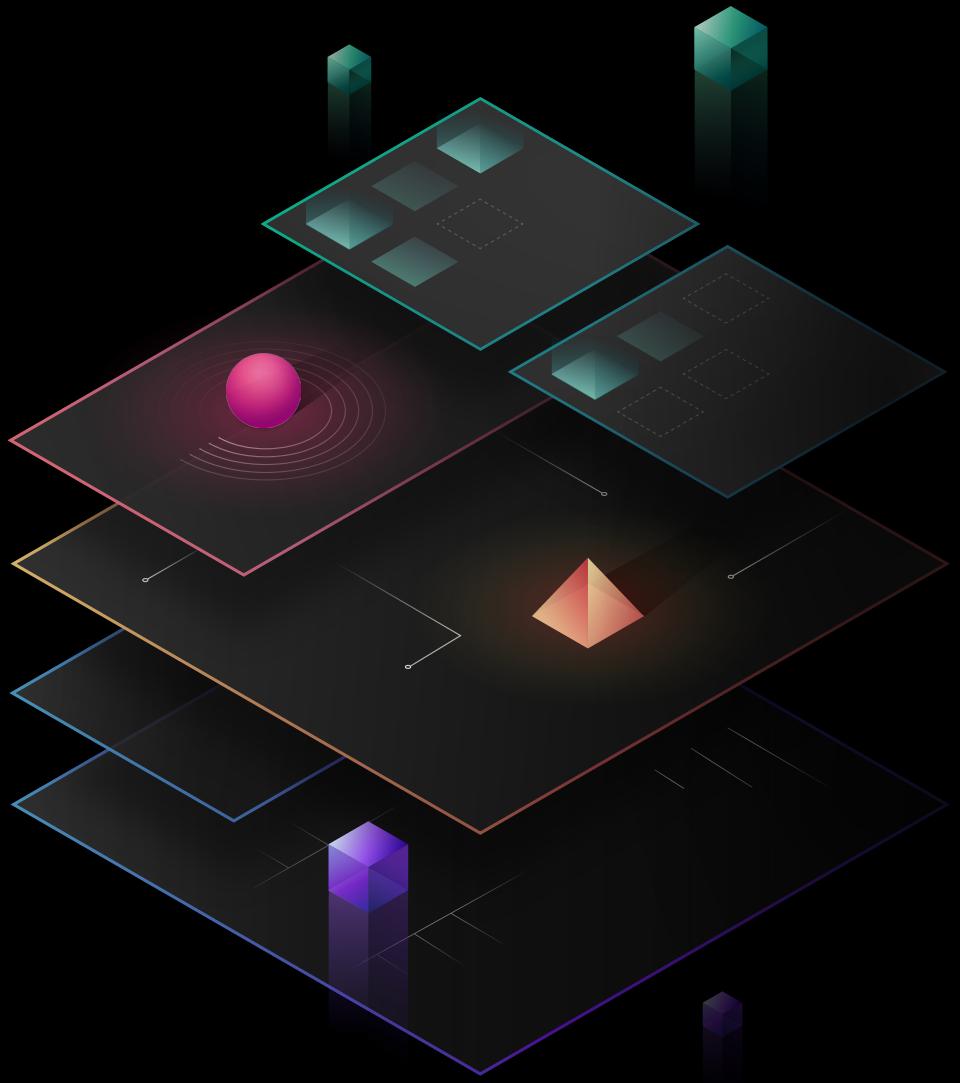




Four Principles of a Cloud Operating Model

The foundations for success in a
multi-cloud world



Contents

<u>Executive Summary</u>	03
<u>Introduction</u>	04
<u>Transitioning to a Multi-Cloud Datacenter</u>	05
<u>Implications of the Cloud Operating Model</u>	08
<u>Unlocking the Cloud Operating Model</u>	09
<u>The 4 Principles of a Cloud Operating Model</u>	10
<u>Step 1: Multi-Cloud Infrastructure Provisioning</u>	10
<u>Step 2: Multi-Cloud Security</u>	13
<u>Step 3: Multi-Cloud Service Networking</u>	16
<u>Step 4: Multi-Cloud Application Delivery</u>	20
<u>Conclusion</u>	23

Executive Summary

To thrive in an era of multi-cloud architecture, driven by digital transformation, enterprise IT must evolve from ITIL-based gatekeeping to enabling shared self-service processes for DevOps excellence.

For most enterprises, digital transformation means delivering new business and customer value more quickly, and at a very large scale. The implication for enterprise IT is a shift from cost optimization to speed optimization. The cloud is an inevitable part of this shift, as it presents the opportunity to rapidly deploy on-demand services with limitless scale.

To unlock the fastest path to value in the cloud, however, enterprises must learn how to industrialize the application delivery process. In practical terms, that means embracing a cloud operating model across each layer of the stack, and implementing corresponding changes to people, process, and tools.

This white paper examines the four principles of a cloud operating model, and presents solutions to help IT teams adopt this model across infrastructure, security, networking, and application delivery.

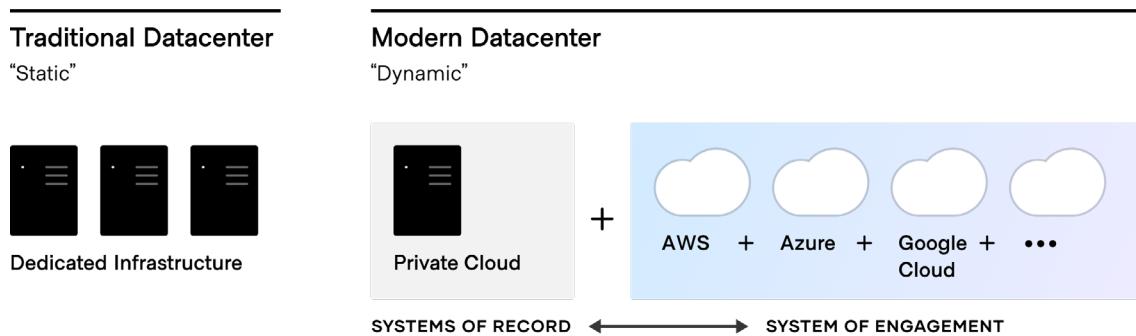
Introduction

A cloud operating model is a new approach for IT operations that organizations need to use to be successful with cloud adoption and thrive in an era of multi-cloud architectures. This overview breaks down the components of that approach and the path to industrializing application delivery across all layers needed to support a cloud-based architecture, articulating the needed changes to people, processes, and tools.

Transitioning to a Multi-Cloud Datacenter

The move to cloud, and multi-cloud, environments is a generational transition for IT. This transition means shifting from largely dedicated servers in a private datacenter to a pool of compute capacity available on demand. While most enterprises began with one cloud provider, there are good reasons to use services from others.

According to HashiCorp's 2021 State of Cloud Strategy Survey, 76% of survey respondents use more than one cloud. And the larger the organization, the more likely they are to use multiple clouds. Inevitably, most Global 2000 organizations are likely to use more than one cloud provider, either by design or due to organic adoption.



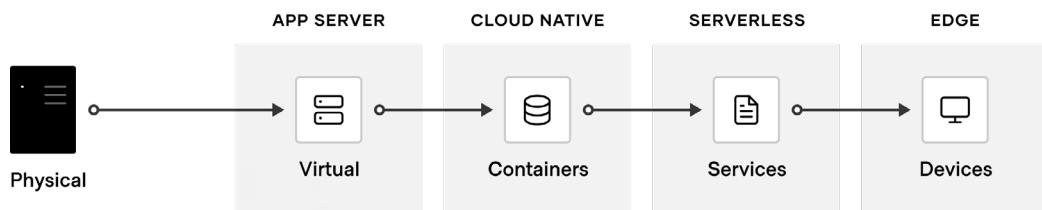
The cloud presents an opportunity for speed and scale optimization for new “systems of engagement” — the applications built to engage customers and users. These new apps are now the primary interface for the customer to engage with a business, and are ideally suited for delivery in the cloud as they tend to:

- Have dynamic usage characteristics, needing to scale loads up and down by orders of magnitude during short time periods.
- Be under pressure to quickly build and iterate. Many of these new systems may be ephemeral in nature, delivering a specific user experience around an event or campaign.

For most enterprises though, these systems of engagement must connect to existing “systems of record” — the core business databases and internal applications, which often continue to reside on infrastructure in existing datacenters. As a result, enterprises typically end up with a hybrid architecture — a mix of multiple public and private cloud environments.

The challenge for most enterprises, then, is how to deliver these applications to the cloud with consistency while also ensuring the least possible friction across the various development teams.

Compounding this challenge, the underlying primitives have changed from manipulating virtual machines in a self-contained environment to manipulating cloud resources in a shared environment. Enterprises then have competing operational models to maintain their existing application estate, while developing the new cloud infrastructure.



For cloud computing to work at its best, enterprises need consistent workflows that can be reused at scale across multiple cloud providers. This requires:

- Consistent instruction sets for provisioning
- Identity for security and for network connections
- Privileges and rights so they can be deployed and run

Implications of the Cloud Operating Model

The essential implication of the transition to the cloud is the shift from “static” infrastructure to “dynamic” infrastructure: from a focus on configuration and management of a static fleet of IT resources to provisioning, securing, connecting, and running dynamic resources on demand.

	Static	Dynamic
 Run	Dedicated Infrastructure	→ Scheduled across the fleet
 Connect	Host-based, Static IP	→ Service-based, Dynamic IP
 Secure	High trust, IP-based	→ Low trust, Identity-based
 Provision	Dedicated servers, Homogeneous	→ Capacity on-demand, Heterogeneous

Decomposing this implication implies several changes of approach. Let's look at each layer of the stack:

- **Provision.** The infrastructure layer transitions from running dedicated servers at limited scale to a dynamic environment where organizations can easily adjust to increased demand by spinning up thousands of servers and scaling them down when not in use. As architectures and services become more distributed, the sheer volume of compute nodes increases significantly.
- **Secure.** The security layer transitions from a fundamentally “high-trust” world enforced by a strong perimeter and firewall to a “low-trust” or “zero-trust” environment with no clear or static perimeter. As a result, the foundational assumption for security shifts from being IP-based to using identity-based access to resources. This shift is highly disruptive to traditional security models.
- **Connect.** The networking layer transitions from being heavily dependent on the physical location and IP address of services and applications to using a dynamic registry of services for discovery, segmentation, and composition. In this environment, enterprise IT teams do not have the same control over the network, or the physical locations of compute resources, and must think about service-based connectivity.
- **Run.** The runtime layer shifts from deploying artifacts to a static application server to deploying applications with a scheduler atop a pool of infrastructure that is provisioned on-demand. In addition, new applications have become collections of services that are dynamically provisioned, and packaged in multiple ways — from virtual machines to containers.

Additionally, each cloud provider offers its own solution to each of these challenges individually. For enterprise IT teams, these shifts in approach are compounded by the realities of running on hybrid- and multi-cloud infrastructures and the varying tools each technology provides.

	Static	Dynamic		
	DEDICATED	PRIVATE CLOUD	AWS	AZURE
 Run Deployment	vSphere	→ vSphere	EKS / ECS Lambda	AKS / ACS Azure Functions
 Connect Networking	Hardware	→ Various Hardware	CloudMap AppMesh	Proprietary
 Secure Security	IP: Hardware	→ Identity: AD/LDAP	Identity: AWS IAM	Identity: Azure AD
 Provision Operations	vCenter	→ Terraform	CloudFormation	Resource Manager
				Cloud Deployment Manager

To address these challenges, teams must ask questions in three key areas:

- **People.** How can we enable a team for a multi-cloud reality, where skills can be applied consistently regardless of target environment?
- **Process.** How do we position central IT services as a self-service enabler of speed versus a ticket-based gatekeeper of control while retaining compliance and governance?
- **Tools.** How do we best unlock the value of the available capabilities of the cloud providers in pursuit of better customer and business value?

Unlocking the Cloud Operating Model

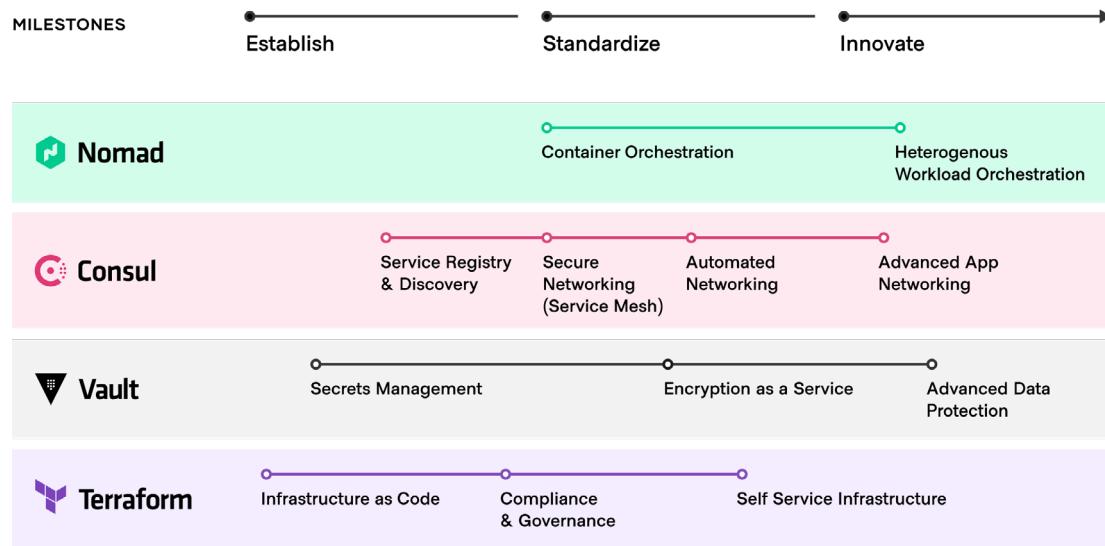
As the implications of a cloud operating model impact teams across infrastructure, security, networking, and applications, we see a repeating pattern amongst enterprises of establishing central shared services — centers of excellence — to deliver the dynamic infrastructure necessary at each layer for successful application delivery.

As teams deliver on each shared service for a cloud operating model, IT velocity increases. The greater cloud maturity an organization has, the faster its velocity.

The typical journey we have seen customers adopt as they unlock a cloud operating model involves three major milestones:

1. **Establish the cloud essentials** – As you begin your journey to the cloud, the immediate requirements are provisioning the cloud infrastructure, typically by adopting infrastructure as code and ensuring it is secure with a secrets management solution. These are the essential foundations for building a scalable, dynamic, futureproof cloud architecture.
2. **Standardize on a set of shared services** – As cloud consumption starts to pick up, you will need to implement and standardize on a set of shared services to take full advantage of the cloud. This introduces challenges around governance and compliance as the need for setting access control rules and tracking requirements become increasingly important.
3. **Innovate using a common logical architecture** – As you fully embrace the cloud and depend on cloud services and applications as the primary systems of engagement, there will be a need to create a common logical architecture. This requires a control plane that connects with the extended ecosystem of cloud solutions and inherently provides advanced security and orchestration across services and multiple clouds.

Example enterprise journey to unlock a Cloud Operating Model



The 4 Principles of a Cloud Operating Model

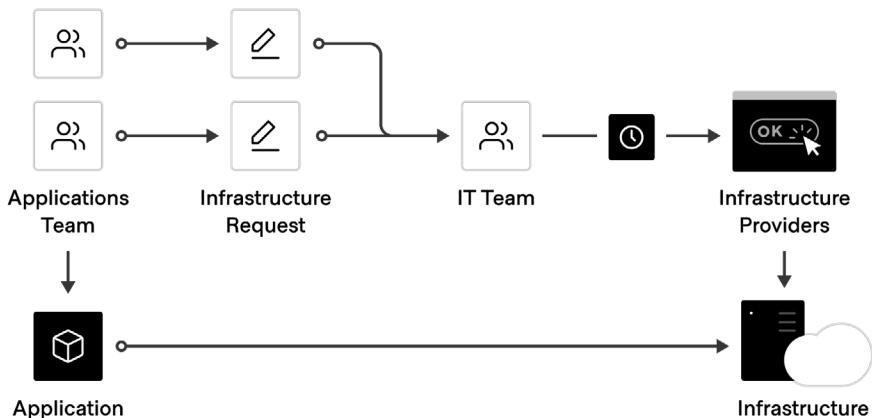
Here are the four foundational principles of a cloud operating model along with what each entails. Each one has its own step-by-step journey that we see organizations consistently use adopt a cloud operating model successfully:

Step 1: Multi-Cloud Infrastructure Provisioning

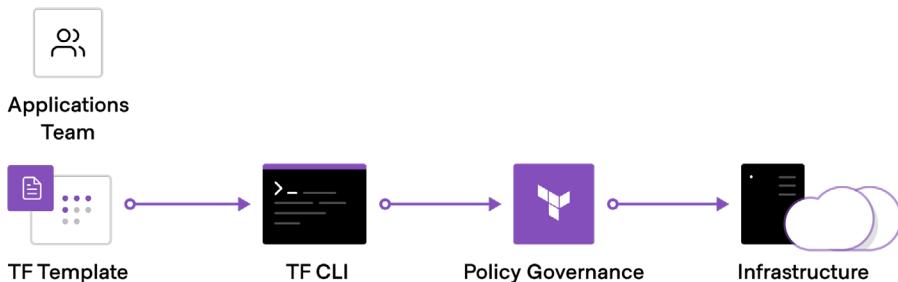
The foundation for adopting the cloud is infrastructure provisioning. [HashiCorp Terraform](#) is the world's most widely used cloud provisioning product and can be used to provision infrastructure for any application using an array of providers for any target platform.

To achieve shared services for infrastructure provisioning, IT teams should start by implementing reproducible infrastructure as code practices, and then layering compliance and governance workflows to ensure appropriate controls.

Before Terraform



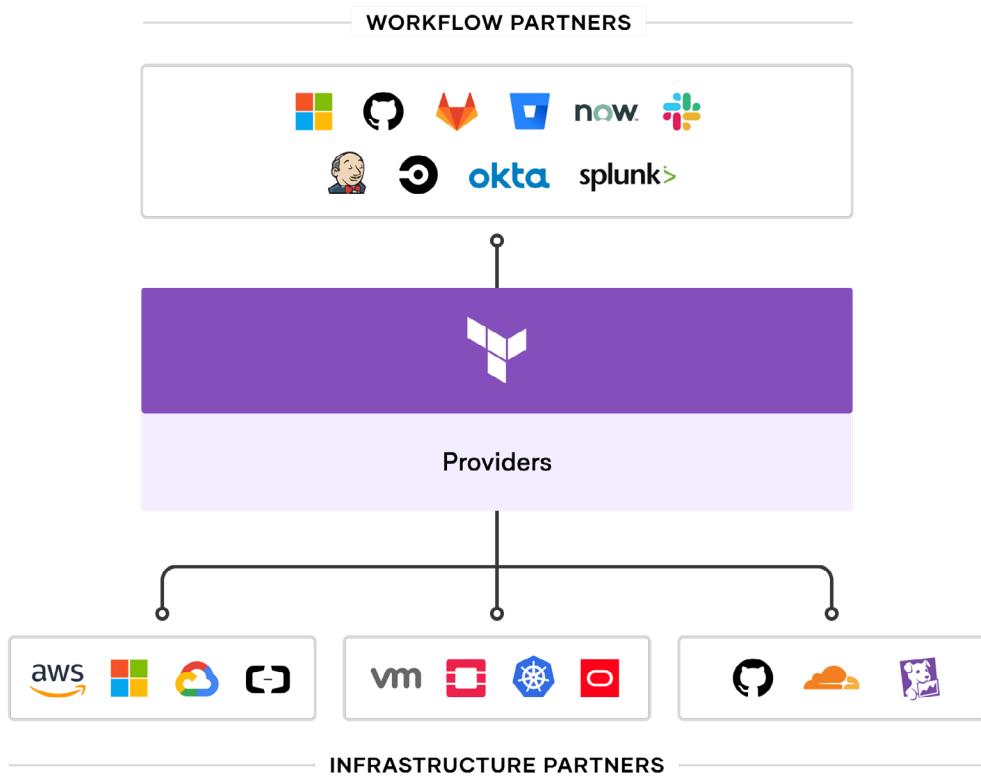
After Terraform



Reproducible Infrastructure as Code

The first goal of a shared service for infrastructure provisioning is to enable the delivery of reproducible infrastructure as code, providing DevOps teams a way to plan and provision resources inside CI/CD workflows using familiar tools throughout.

DevOps teams can create Terraform templates that express the configuration of services from one or more cloud platforms. Terraform integrates with all major configuration management tools to allow fine-grained provisioning to be handled following the provisioning of the underlying resources. Finally, templates can be extended with services from many other ISV providers to include monitoring agents, application performance monitoring (APM) systems, security tooling, DNS, and content delivery networks, and more. Once defined, the templates can be provisioned as required in an automated way. In doing so, Terraform becomes the lingua franca and common workflow for teams provisioning resources across public and private cloud.



For self-service IT, the decoupling of the template-creation process and the provisioning process greatly reduces the time taken for any application to go live since developers no longer need to wait for operations approval, as long as they use a pre-approved template.

Compliance and Management

Most teams also need to enforce policies on the type of infrastructure created, how it is used, and which teams get to use it. HashiCorp's Sentinel policy as code framework provides compliance and governance without requiring a shift in the overall team workflow, and is defined as code too, enabling collaboration and comprehension for DevSecOps.

Without policy as code, organizations resort to using a ticket-based review process to approve changes. This can result in developers waiting weeks or longer to provision infrastructure and becomes a bottleneck. Policy as code solves this by splitting the definition of the policy from the execution of the policy.

Centralized teams codify policies enforcing security, compliance, and operational best practices across all cloud provisioning. Automated enforcement of policies ensures changes are in compliance without creating a manual review bottleneck.

Step 2: Multi-Cloud Security

Dynamic cloud infrastructure means a shift from host-based identity to application-based identity, with low- or zero-trust networks across multiple clouds without a clear network perimeter.

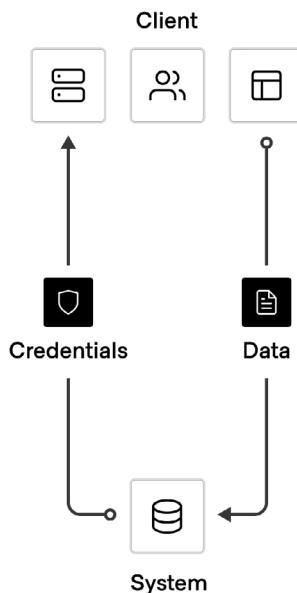
In the traditional security world, we assumed high trust internal networks, which resulted in a hard shell and soft interior. With the modern “zero trust” approach, we work to harden the inside as well. This requires that applications be explicitly authenticated, authorized to fetch secrets and perform sensitive operations, and tightly audited.

HashiCorp Vault enables teams to securely store and tightly control access to tokens, passwords, certificates, and encryption keys for protecting machines and applications. This provides a comprehensive secrets management solution. Beyond that, Vault helps protect data at rest and data in transit. Vault exposes a high-level API for cryptography for developers to secure sensitive data without exposing encryption keys. Vault also can act like a certificate authority, to provide dynamic short lived certificates to secure communications with SSL/TLS. Lastly, Vault enables a brokering of identity between different platforms, such as Active Directory on premises and AWS Identity and Access Management (IAM) to allow applications to work across platform boundaries.

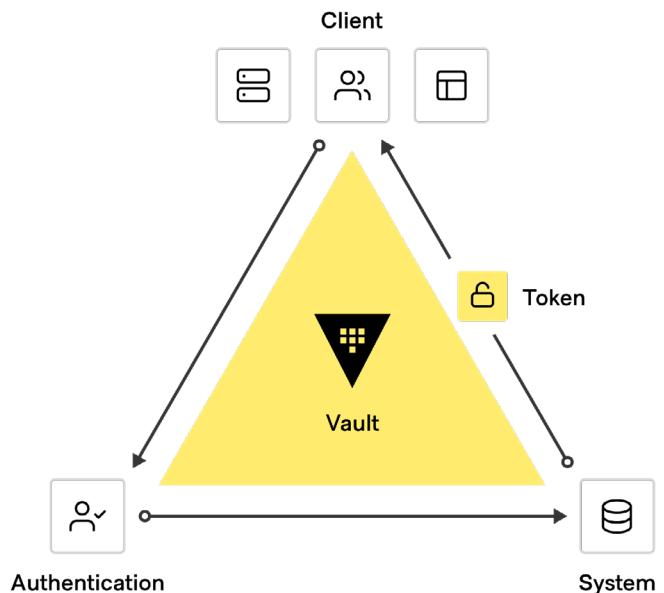
Vault is widely used across many industries including stock exchanges, large financial organizations, and hotel chains to provide security in a cloud operating model.

To achieve shared services for security, IT teams should enable centralized secrets management services, and then use that service to deliver more sophisticated Encryption-as-a-Service use cases such as certificate and key rotations, and encryption of data in transit and at rest.

Before Vault



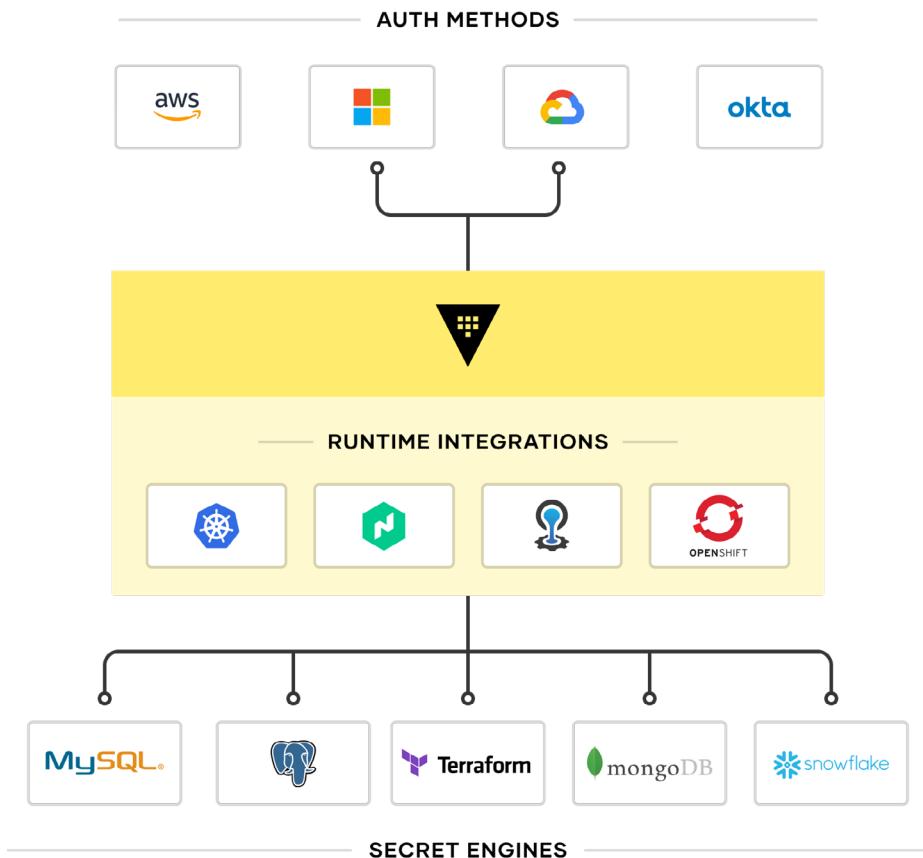
After Vault



Secrets Management

The first step in cloud security is typically secrets management: the central storage, access control, and distribution of dynamic secrets. Instead of depending on static IP addresses, integrating with identity-based access systems such as AWS IAM and Azure AAD to authenticate and access services and resources is crucial.

Vault uses policies to codify how applications authenticate, which credentials they are authorized to use, and how auditing should be performed. It can integrate with an array of trusted identity providers such as cloud identity and access management platforms, Kubernetes, Active Directory, and other SAML-based systems for authentication. Vault then centrally manages and enforces access to secrets and systems based on trusted sources of application and user identity.



Enterprise IT teams should build a shared service that enables the request of secrets for any system through a consistent, audited, and secured workflow.

Encryption-as-a-Service

Additionally, enterprises need to encrypt application data at rest and in transit. Vault can provide Encryption-as-a-Service to provide a consistent API for key management and cryptography. This allows developers to perform a single integration and then protect data across multiple environments.

Using Vault as a basis for Encryption-as-a-Service solves difficult problems faced by security teams, such as certificate and key rotation. Vault enables centralized key management to simplify encrypting data in transit and at rest across clouds and datacenters. This helps reduce costs around expensive hardware security modules (HSMs) and increases productivity with consistent security workflows and cryptographic standards across the organization.

While many organizations provide a mandate for developers to encrypt data, they don't often provide the "how", which leaves developers to build custom solutions without an adequate understanding of cryptography. Vault provides developers a simple API that can be easily used, while giving central security teams the policy controls and lifecycle management APIs they need.

Advanced Data Protection

Organizations moving to the cloud or spanning hybrid environments still maintain and support on-premises services and applications that need to perform cryptographic operations, such as data encryption for storage at rest. These services do not necessarily want to implement the logic around managing these cryptographic keys, and thus seek to delegate the task of key management to external providers. Advanced Data Protection allows organizations to securely connect, control, and integrate advanced encryption keys, operations, and management between infrastructure and Vault Enterprise, including automatically protecting data in MySQL, MongoDB, PostgreSQL, and other databases using transparent data encryption (TDE).

For organizations that have high security requirements for data compliance (PCI DSS, HIPAA, etc), protecting data, and cryptographically-protecting anonymity for personally identifiable information (or PII), Advanced Data Protection provides organizations with functionality for data tokenization, such as data masking, to protect sensitive data, such as credit cards, sensitive personal information, bank numbers, etc.

Step 3: Multi-Cloud Service Networking

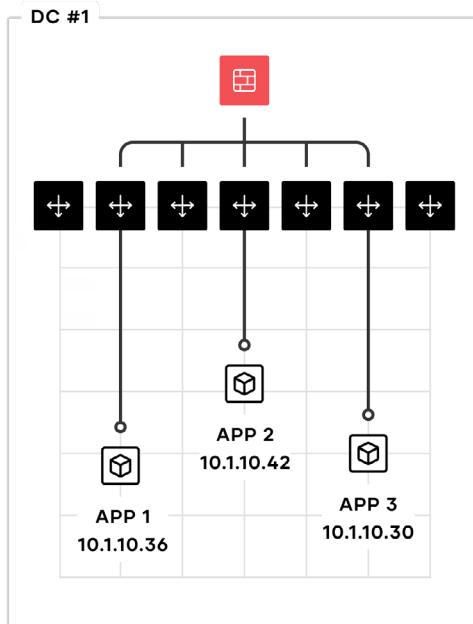
The challenges of networking in the cloud can be some of the most difficult aspects of adopting the cloud operating model for enterprises. The combination of dynamic IP addresses, a significant growth in east-west traffic as the microservices pattern is adopted, and the lack of a clear network perimeter present formidable obstacles.

[HashiCorp Consul](#) provides a multi-cloud service networking layer to connect and secure services. Consul is a widely deployed product, with many customers running significantly more than 100,000 nodes in their environments.

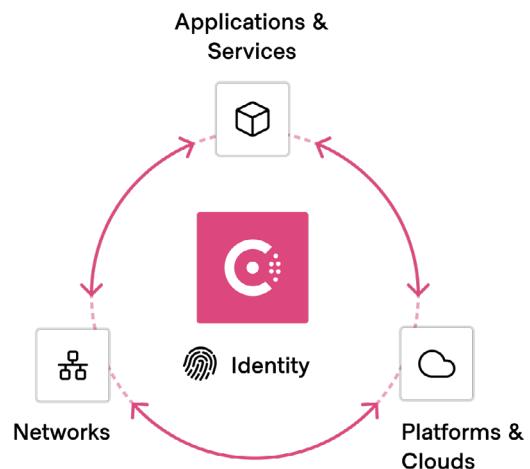
Networking services should be provided centrally, whereby IT teams provide service registry and service discovery capabilities. Having a common registry provides a "map" of what services are running, where they are, and their current health status. The registry can be queried programmatically to enable service discovery or drive network automation of API gateways, load balancers, firewalls, and other critical middleware components. These middleware components can be moved out of the network

by using a service mesh approach, where proxies run on the edge to provide equivalent functionality. Service mesh approaches allow the network topology to be simplified, especially for multi-cloud and multi-datacenter topologies

Before Consul



After Consul



Service Discovery

The starting point for networking in a cloud operating model is typically a common service registry, which provides a real-time directory of what services are running, where they are, and their current health status. Traditional approaches to networking rely on load balancers and virtual IPs to provide a naming abstraction to represent a service with a static IP. The process to track the network location of services often takes the form of spreadsheets, load balancer dashboards, or configuration files, all of which are disjointed, manual processes that are not ideal.

For Consul, each service is programmatically registered and DNS and API interfaces are provided to enable any service to be discovered by other services. The integrated health check will monitor each service instance's health status so the IT team can triage the availability of each instance and Consul can help prevent routing traffic to unhealthy service instances. Consul can be integrated with other

services that manage existing north-south traffic such as traditional load balancers and distributed application platforms such as Kubernetes, to provide a consistent registry and discovery service across multiple runtimes and clouds.

Network Infrastructure Automation

The next step is to reduce the operational complexity of existing networking infrastructure through network automation. Instead of a manual, ticket-based process to reconfigure load balancers and apply firewall rules every time there is a change in service network locations or configurations, Consul can automate these network operations using Terraform to execute changes based on predefined tasks. This is achieved by enabling network infrastructure devices to subscribe to service changes from the service registry, creating highly dynamic infrastructure that can scale significantly higher than static-based approaches.

This configuration — whereby Terraform executes configurations based on event changes detected by Consul — removes dependencies and obstacles for common tasks. Product teams can independently deploy applications while IT teams can rely on Consul to handle the downstream automation those product teams require. The benefits persist throughout the lifecycle of the service: automation can properly close firewall ports as services are retired.

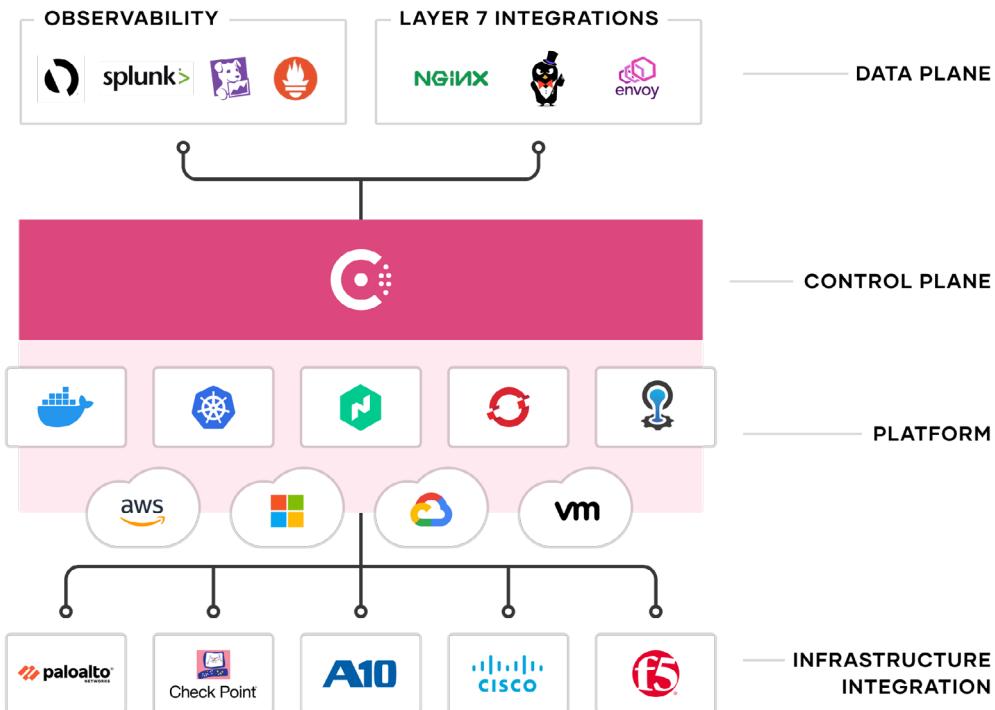
Zero Trust Networking with Service Mesh and API Gateway

As organizations continue to scale with microservices-based or cloud native applications, the underlying infrastructure becomes larger and more dynamic with an explosion of east-west traffic. This causes a proliferation of expensive network middleware devices with single points of failure and significant operational overhead exposed to IT teams.

Consul provides a distributed service mesh that pushes routing, authorization, and other networking functionalities to the endpoints in the network, rather than imposing them through middleware. This makes the network topology simpler and easier to manage, reduces the need for expensive middleware devices within east-west traffic paths, and makes service-to-service communication more reliable and scalable. Adding Consul's API Gateway provides consistent control and security for how north-south traffic is handled through a single, centralized control plane.

Consul is an API-driven control plane that integrates with sidecar proxies alongside each service instance (proxies such as Envoy). These proxies provide the distributed data plane. Together, these two planes enable a zero trust network model that ensures all service-to-service communication is

authenticated, authorized, and encrypted. This security posture is achieved with automatic mutual TLS encryption and identity-based authorization. Operations teams can work with the appropriate stakeholders to define security policies with logical services (rather than IP addresses) and provide least-privilege access to developers.



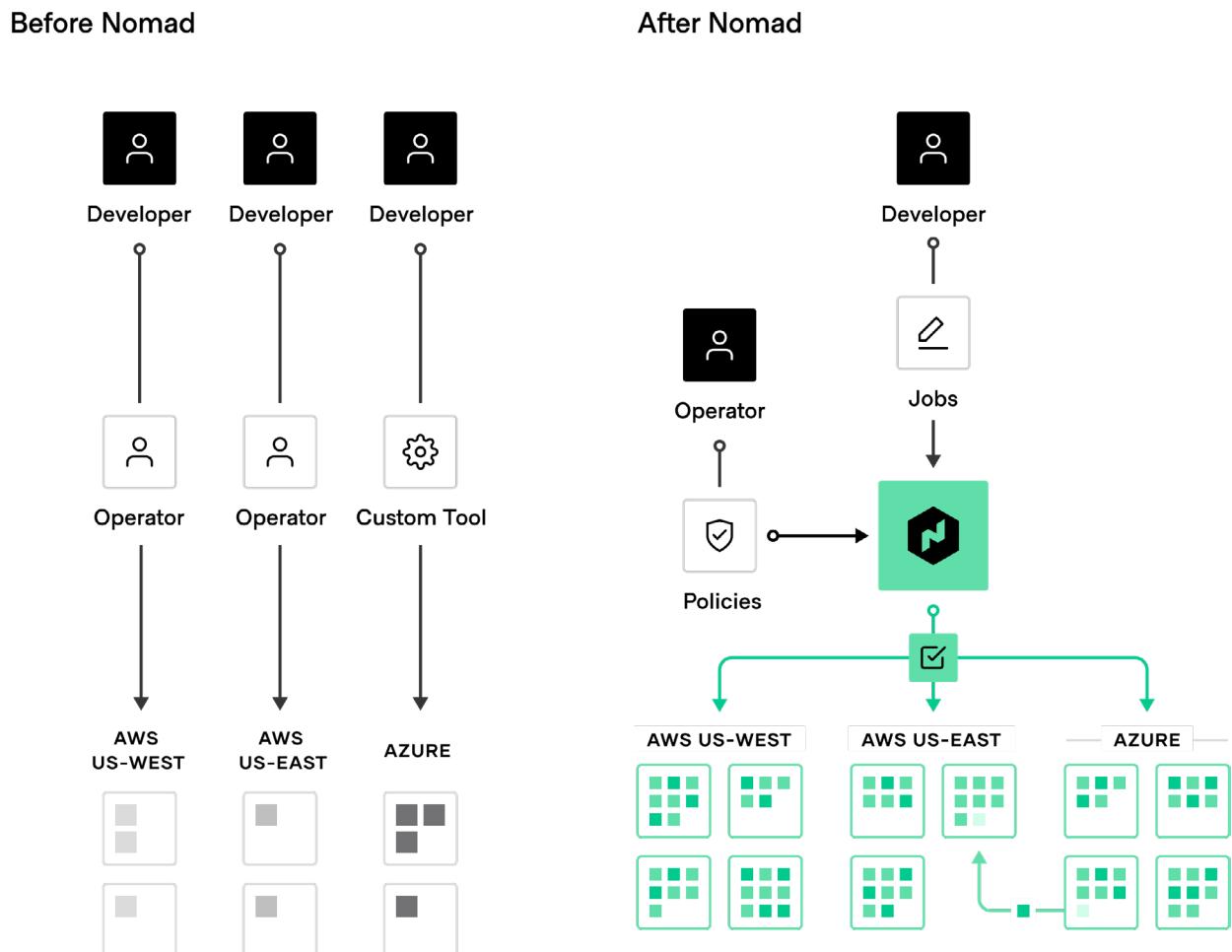
Consul can be integrated with Vault for centralized PKI and certificate management with automatic certificate rotation on both the control plane and data plane. Consul's integration also extends to Kubernetes deployments, including the storage of sensitive data such as keys and tokens in Vault. This approach reduces risk compared to relying on native Kubernetes secrets.

Consul Service Mesh secures service connections across any cloud environment, and on any runtime. This consistent dataplane allows developers and IT teams to connect their services across heterogeneous environments and abstractions. Furthermore, Consul supports multi-tenancy with Administrative Partitions. With this feature, multiple deployments can remain under a single control plane allowing for consistent management and governance while maintaining autonomy and isolation for different tenants.

Step 4: Multi-Cloud Application Delivery

Finally, at the application layer, new apps are increasingly distributed while legacy apps also need to be managed more flexibly. [HashiCorp Nomad](#) provides a flexible orchestrator to deploy and manage legacy and modern applications, for all types of workloads: from long running services, to short lived batch, to system agents.

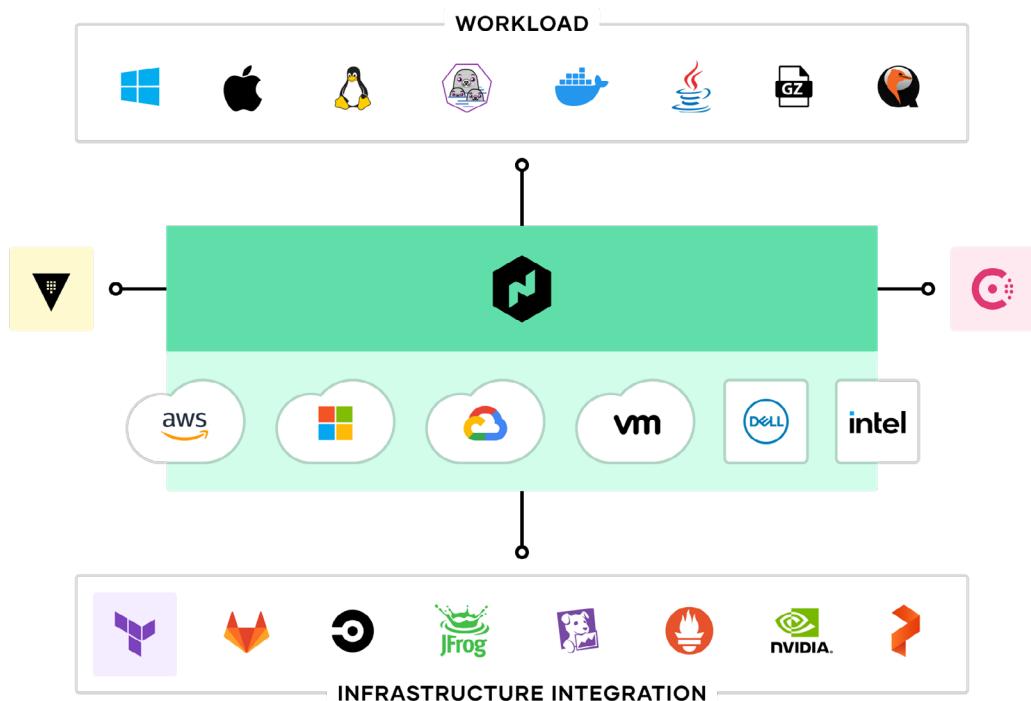
To achieve shared services for application delivery, IT teams should use Nomad in concert with Terraform, Vault, and Consul to enable the consistent delivery of applications on cloud infrastructure, incorporating necessary compliance, security, and networking requirements, as well as workload orchestration and scheduling.



Mixed Workload Orchestration

Many new workloads are developed with container packaging with the intent to deploy to Kubernetes or other container-management platforms. But many heritage workloads will not be moved onto those platforms, nor will future serverless applications. Nomad provides a consistent process for deployment of all workloads from virtual machines through standalone binaries and containers, and provides core orchestration benefits across those workloads such as release automation, multiple upgrade strategies, bin packing, and resilience.

For modern applications — typically built in containers — Nomad provides the same consistent workflow at scale in any environment. Nomad is focused on simplicity and effectiveness at orchestration and scheduling, and avoids the complexity of platforms such as Kubernetes that require specialist skills to operate and solve only for container workloads.



Nomad integrates into existing CI/CD workflows to provide fast, automatic application deployments for legacy and modern workloads.

High Performance Compute

Nomad is designed to schedule applications with low latency across very large clusters. This is critical for customers with large batch jobs, as is common with high performance computing (HPC) workloads. In the [Two Million Container Challenge](#), Nomad was able to schedule two million Docker containers on 6,100 hosts in 10 AWS regions in 22 minutes. Several enterprise Nomad deployments run at even larger scales.

Nomad enables high-performance applications to easily use an API to consume capacity dynamically, enabling efficient sharing of resources for data-analytics applications like Spark. The low-latency scheduling ensures results are available in a timely manner and minimizes wasted idle resources.

Multi-Datacenter Workload Orchestration

Nomad is multi-region and multi-cloud by design, with a consistent workflow when deploying any workload. As teams roll out global applications in multiple datacenters or across cloud boundaries, Nomad provides orchestration and scheduling for those applications, supported by the infrastructure, security, and networking resources and policies to ensure the application is successfully deployed.

Conclusion

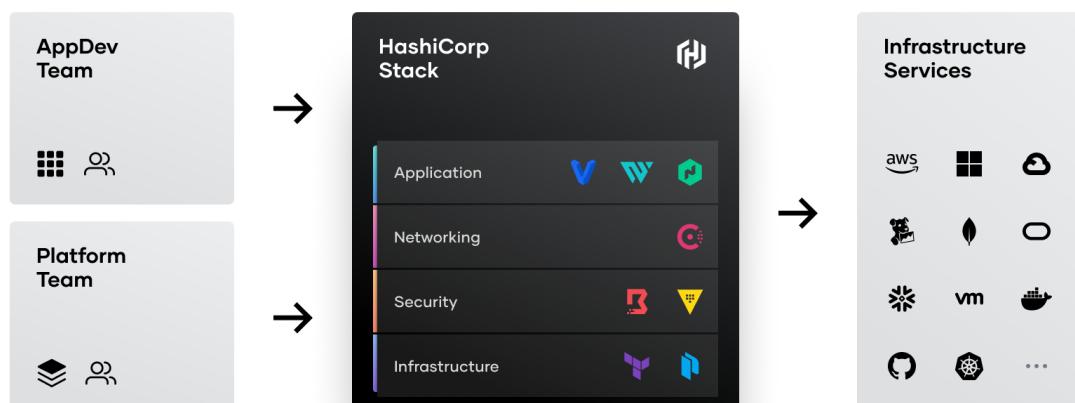
Ultimately, these shared services across infrastructure, security, networking, and application runtime present an industrialized process for application delivery, all while taking advantage of the dynamic nature of each layer of the cloud.

Embracing a cloud operating model enables self-service IT that is fully compliant and governed for teams to deliver applications at increasing speed.

Transitioning to a cloud operating model is an inevitable shift for enterprises aiming to maximize their digital transformation efforts. The HashiCorp suite of tools seeks to provide solutions for each layer of the cloud – and each principle of the cloud operating model – to enable enterprises to modernize their operations.

Enterprise IT needs to evolve away from ITIL-based control points, with their focus on cost optimization, toward becoming self-service enablers focused on speed optimization. It can do this by delivering shared services across each layer of the cloud, designed to assist teams deliver new business and customer value at speed.

Unlocking the Cloud Operating Model



Unlocking the fastest path to value in a modern multi-cloud data center through adopting a common cloud operating model means shifting characteristics of enterprise IT across all three key areas:

- **People: Shifting to multi-cloud skills**

- Reuse skills from internal datacenter management and single cloud vendors and apply them consistently in any environment.
- Embrace DevSecOps and other agile practices to continuously deliver increasingly ephemeral and distributed systems.

- **Process: Shifting to self-service IT**

- Position central IT as an enabling shared service focused on application delivery velocity: shipping software ever more rapidly with minimal risk.
- Establish centers of excellence across each layer of the cloud for self-service delivery of capabilities.

- **Tools: Shifting to dynamic environments**

- Use tools that support the increasing ephemerality and distribution of infrastructure and applications and that support the critical workflows rather than being tied to specific technologies.
- Provide policy and governance tooling to match the speed of delivery with compliance to manage risk in a self-service environment.

About HashiCorp

HashiCorp is a leader in multi-cloud infrastructure automation software. The HashiCorp software suite enables organizations to adopt consistent workflows and create a system of record for automating the cloud: infrastructure provisioning, security, networking, and application deployment. HashiCorp's portfolio of products includes Vagrant™, Packer™, Terraform®, Vault™, Consul, Nomad™, Boundary, and Waypoint™. HashiCorp offers products as open source, enterprise, and as managed cloud services. The company is headquartered in San Francisco, though most of HashiCorp employees work remotely, strategically distributed around the globe. For more information, visit hashicorp.com or follow HashiCorp on Twitter [@HashiCorp](https://twitter.com/HashiCorp).

