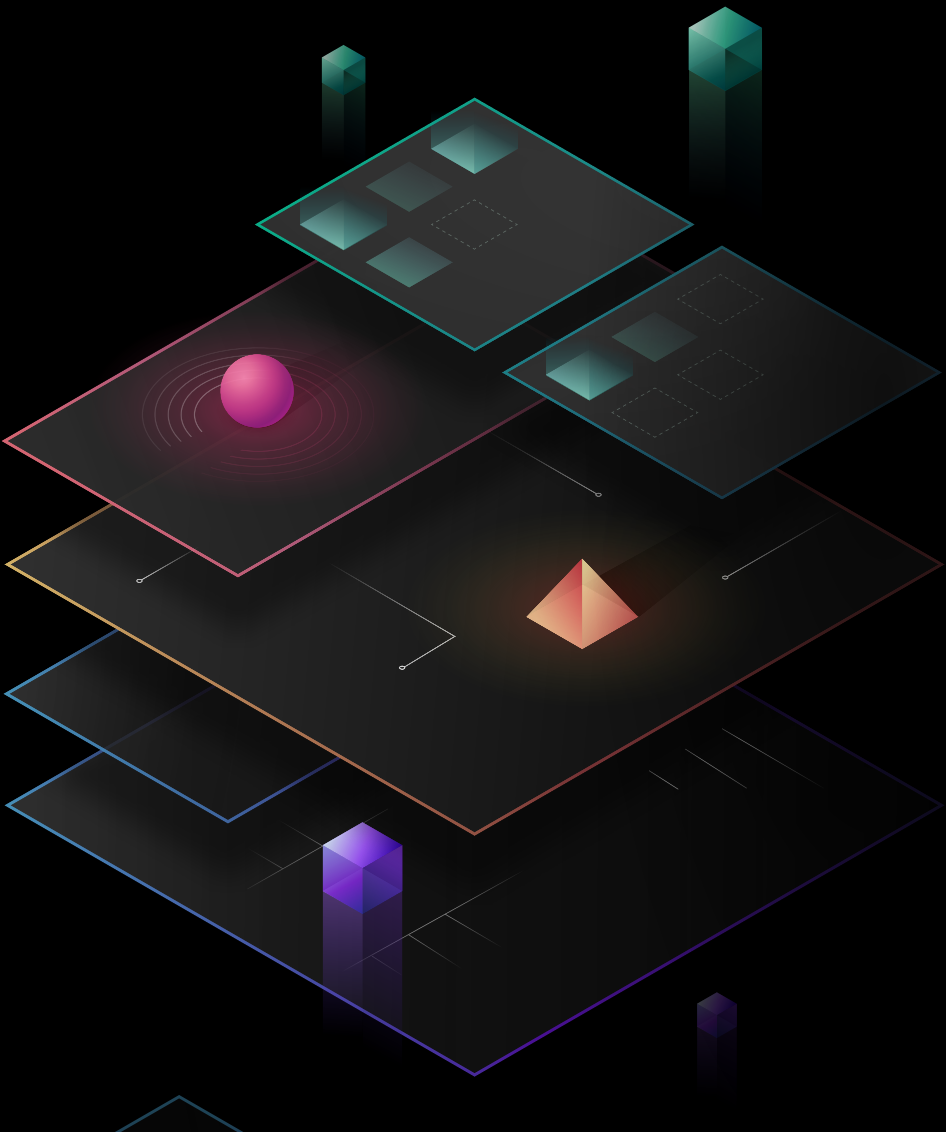
HashiCorp

Enabling a Cloud Operating Model

A modern approach to provision, secure, connect, and run your cloud resources



WHITEPAPER

Contents

Executive Summary

The cloud demands a new operating model. Datacenter primitives are changing: static to dynamic infrastructure, security and networking policies based on identity rather than IP addresses, and an automated self-service platform as opposed to manual ticketing systems. While this transformation speeds development and shrinks go-to-market times, it renders much of the previous generation of tools and processes obsolete.
To fully realize the benefits of cloud computing, enterprises must transition to scalable dynamic workflows and management at every cloud layer. From this they can deploy shared services enabled by platform teams to improve speed, increase efficiency, and decrease risk.

For most enterprises, digital transformation means delivering new business and customer value faster, at a very large scale. The implication for a cloud strategy/program office, then, is to balance the need for developer speed with cost optimization and risk mitigation. The cloud is an inevitable part of this shift as it presents the opportunity to rapidly deploy on-demand services with limitless scale. This re-platforming, driven by cloud services, requires new models for architecting applications and completely new approaches to infrastructure provisioning, security, networking, and deployment.

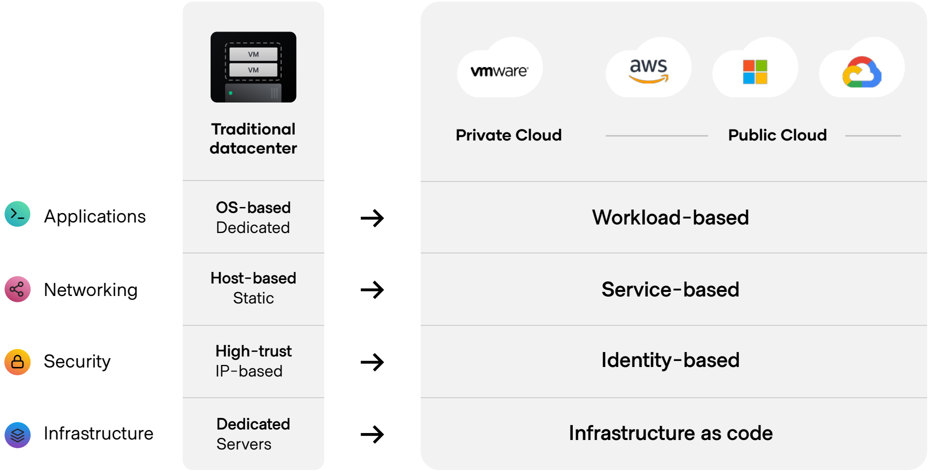
In this white paper, we look at the value of a cloud operating model across the cloud layers of infrastructure, security, networking, and applications, and address the role that platform teams play in making this model a reality.

A new cloud operating model

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.

Cloud computing is a generational transition, shifting from largely static, dedicated servers in a private datacenter to a pool of service capacity available on demand from a variety of different providers. Successfully adopting the cloud means enabling a cloud operating model to address the transitional concerns at each operational layer: infrastructure, security, networking, and applications.

- For infrastructure, provisioning and management is done using infrastructure as code
- For security, brokering access to and management of sensitive data is based on identity
- For networking, access and connections are based on service identity
- For applications, deployment and management is workload-based



	Traditional datacenter		Private Cloud — Public Cloud
Applications	OS-based Dedicated	→	Workload-based
Networking	Host-based Static	→	Service-based
Security	High-trust IP-based	→	Identity-based
Infrastructure	Dedicated Servers	→	Infrastructure as code

As infrastructure, security, networking, and applications teams move from traditional datacenters to the cloud, the foundations of each layer change.

For each layer the goal is to build a consistent system of engagement for developers and a system of record for the platform team — the team of cloud engineers responsible for implementing a cloud operating model as a platform of standardized shared tools and services.

Challenges of cloud adoption

With a transition to the cloud, organizations find themselves navigating the complexity of multi-cloud and a variety of different tooling and processes connected to each platform. Organizations bogged down in this tactical approach to their cloud adoption often struggle with inefficiency, security and compliance risk, and runaway costs from isolated teams and disconnected tools.

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

- **People:** How can we enable a team for a cloud and/or multi-cloud reality, where skills can be applied consistently regardless of target environment or development team?
- **Process:** How do we position the cloud strategy/program office as a self-service enabler of speed instead of a ticket-based gatekeeper of control, while retaining compliance and governance across all four cloud layers?
- **Tools:** How do we best unlock the value of the cloud through tools and automation that enable development teams to provide better customer and business value while minimizing risk?

The answer to these questions is a cloud operating model that enables standardized shared services at all layers of the cloud.

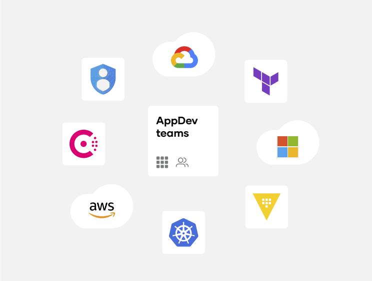
Successfully enabling a cloud operating model

Organizations that successfully enable a cloud operating model follow a typical blueprint called a maturity model and rely on centralized cloud platform teams.

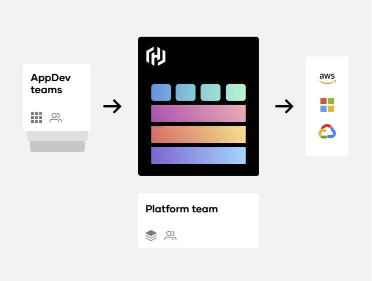
Cloud maturity model

Cloud adoption journeys typically follow an established pattern that flows through three stages:

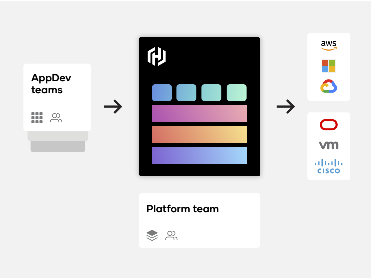
- **Stage 1: Adopting** — This is the beginning of cloud experimentation and usage, defined by individual teams engaging with cloud providers in silos to deliver applications and services. This leads to multiple different workflows, each best suited to a particular team's needs. At this stage there is no platform, limited knowledge share and minimal cloud operating strategy. This is what we refer to as a tactical cloud.
- **Stage 2: Standardizing** — As cloud usage increases, the organization shifts to strategic planning by a platform team to standardize the way developers interface with the cloud. The platform team is tasked with creating central services around provisioning, security, networking, and application deployment. This process accelerates developer productivity by removing the manual tasks associated with deploying cloud resources. At the same time, it reduces risk by providing a centralized way to apply corporate governance and security policies to all cloud-based resources. From here, teams adopt a cloud operating model resulting in the establishment of a cloud program.
- **Stage 3: Scaling** — Once established in a single cloud environment, platform teams can extend these workflows to other cloud vendors and across an organization's private estate, creating a consistent platform and system across all development and deployment areas. The team implements enterprise-scale solutions that can facilitate self-service cloud workflows across dozens or hundreds of teams.



Stage 1: Adopting



Stage 2: Standardizing



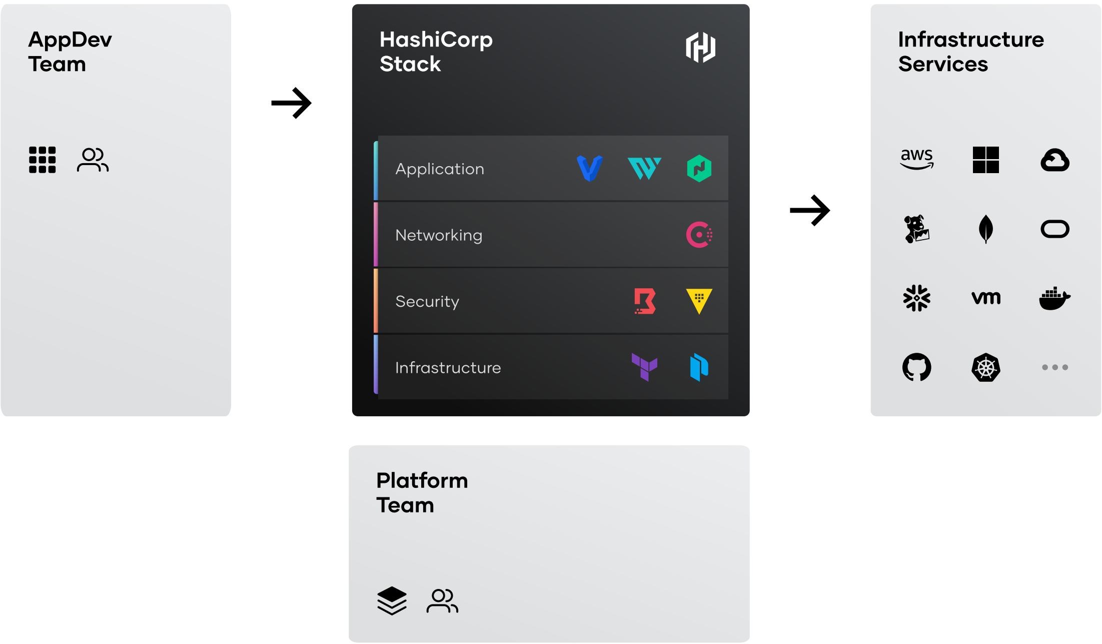
Stage 3: Scaling

Platform teams accelerate cloud adoption

The platform team is responsible for delivering each layer of a cloud operating model as a standardized shared service that can be consumed by end-users in the organizations. This model underpins some of the largest organizations that have successfully transitioned to the cloud, including

In addition to providing standardized shared services, the platform team:

- **Establishes best practices for developers and practitioners** to engage with cloud services. These best practices are then codified directly into workflows and educational programs such as Cloud Centers of Excellence (CCoEs).
- **Serves as a single point of integration for other corporate teams**, including security, compliance, financial controls, etc. to ensure that their best practices are baked into relevant workflows.
- **Provides a central system for tracking, reporting, and auditing** to inform adoption and usage of the cloud operating model. This informs progress of the organization and highlights areas where there may be risk exposure: such as security, compliance, and spending.



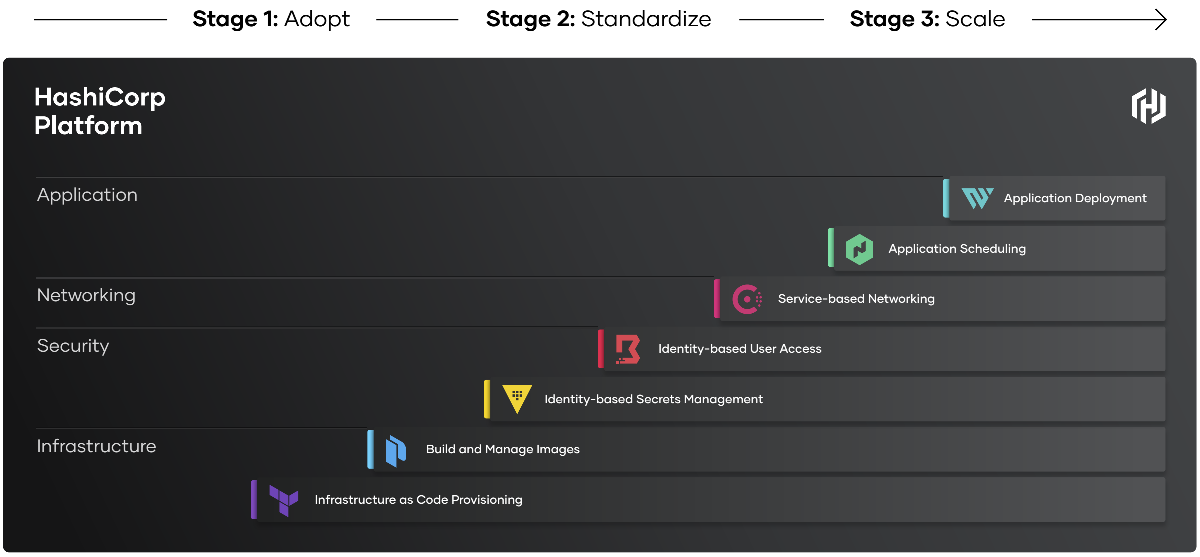
Enterprise platform capabilities

The key to implementing a cloud operating model is the creation of consistent workflows and processes. These drive the delivery of core capabilities organizations need for their cloud platform. We have identified six functional areas that need to be built into the cloud platform:

1. **Unified workflow management:** Implement central leadership and processes to unify common workflows across all cloud layers and teams.
2. **Reliability and scale:** Create solutions for the cloud platform that are dependable and perform consistently at scale across all levels of an organization.
3. **Policy and security:** Incorporate tools to enable the integration of policies and guardrails directly into your workflows and cloud platform.
4. **Governance, risk, and compliance:** Establish a consistent philosophy for the integration of security and compliance frameworks directly into all layers of the cloud platform.
5. **Visibility and optimization:** Build tools and dashboards to view and audit all aspects of your cloud platform to ensure consistent performance and drive optimization.
6. **Integration and API-driven workflows:** Create tooling and integrations to ensure the platform has the functionality required and can be easily adopted by the organization.

One cloud infrastructure automation platform

HashiCorp provides a platform of products for cloud infrastructure automation to enable a cloud operating model. HashiCorp products are aligned to each layer of a cloud operating model (infrastructure, security, networking, and application) and their corresponding use cases:



Learn more about each layer and the corresponding use cases in the following sections.

Infrastructure layer

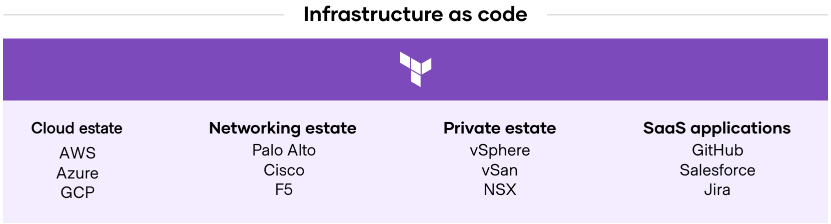
As organizations move from on-premises infrastructure to cloud infrastructure, operators face new challenges:

- **Scale:** Teams want to quickly scale their infrastructure usage up and down with no errors despite potentially extensive configuration changes.
- **Variety:** Teams want unified provisioning workflows on a variety of platforms.
- **Dependencies:** Teams want to include and automate existing services and dependencies into configurations as part of this provisioning workflow.

The foundational layer of a cloud operating model is infrastructure provisioning using an infrastructure as code (IaC) approach. Converting infrastructure into code allows teams to declaratively define the desired end state of their infrastructure, ensuring consistency in every deployment, while also allowing them to track and audit changes to that code.

Standardize infrastructure provisioning with HashiCorp Terraform

HashiCorp Terraform enables this IaC provisioning of infrastructure. With a fully extensible engine, it has thousands of pre-built integrations with cloud providers and popular software applications to make it the central tool for all infrastructure provisioning use cases:



Infrastructure as code

Cloud estate	Networking estate	Private estate	SaaS applications
AWS Azure GCP	Palo Alto Cisco F5	vSphere vSan NSX	GitHub Salesforce Jira

Typically, organizations start with a single cloud service provider, then extend to other clouds. From there, they begin deploying networking tools. Then they are ready to expand the cloud operating model back into their private cloud environments and SaaS applications.

- **Cloud estate:** From a single cloud to multiple, including all major public clouds
- **Networking estate:** Integration of all major networking platforms
- **Private estate:** Expansion of workflows to private datacenters
- **SaaS applications:** Expand capabilities with integrations from an extensive library of providers and integrations

By creating a shared service for infrastructure provisioning, Terraform provides product teams a way to plan and provision all of these resource types inside CI/CD workflows using familiar tools. That makes Terraform the lingua franca and common workflow for teams provisioning resources to help their platform scale and extend its capabilities.

Terraform takes a declarative approach to configuration, which asserts the desired state of a set of infrastructure and can be used to deliver and manage that state across an entire organization. As organizations mature in their cloud journey, Terraform supports more advanced services that enable it to be run as a shared service to facilitate all infrastructure across an organization's varied hybrid estates.

These shared services allow platform teams to deploy common workflows to fit their core needs, including:

- **Unified workflow management and integrations:** Teams can increase productivity with change tracking and versioning, reusability, and centralized configuration. They can use the same workflow to manage multiple cloud providers and handle cross-cloud dependencies with 2,500+ public providers in the Terraform Registry. And they can extend Terraform's automation with CI/CD integrations, API access, and third-party services with run tasks.
- **Reliability and scale:** No-code provisioning lets platform teams manage a catalog of no-code modules for users such as app developers to deploy without prior Terraform experience. They can publish a library of approved configurations in a private internal module registry for use across their organization and natively integrate with public registries like ServiceNow Service Catalog to extend existing self-service processes.
- **Policy, security, and compliance:** Platform teams can reduce the risk of misconfigurations by enforcing guardrails for security, compliance, cost, and organizational best practices before infrastructure is provisioned. This is done with a combination of Sentinel (HashiCorp's policy as code framework), other policy frameworks like Open Policy Agent (OPA), and third-party checks integrated through Terraform run tasks to establish and automatically enforce critical policies across all infrastructure.
- **Visibility and optimization:** Organizations can gain instant visibility into the state of cloud infrastructure with Terraform health assessments. Drift Detection for Terraform preemptively detects when a resource has changed from the expected state, helping reduce security and operational risks. Customizable notifications alert operators when drift is detected or a health assessment fails. Teams can also minimize cloud waste and gain visibility into the impact of changes with cost estimation.

Build image pipelines with HashiCorp Packer

The next step is for organizations to standardize their cloud infrastructure built by Terraform through the creation of "golden images." These identical system and container images for multiple clouds are built with HashiCorp Packer from a single source template. This ensures consistency across development teams and speeds up developer onboarding and application deployment. Additionally, teams that desire more automation can use HCP Packer, a cloud service that builds a registry of these images. Once built, the images in this registry can easily be shared and then deployed by Terraform. This toolchain lets security and platform teams work together to create, manage, and consume images in a centralized way, with an abstraction layer for developers.

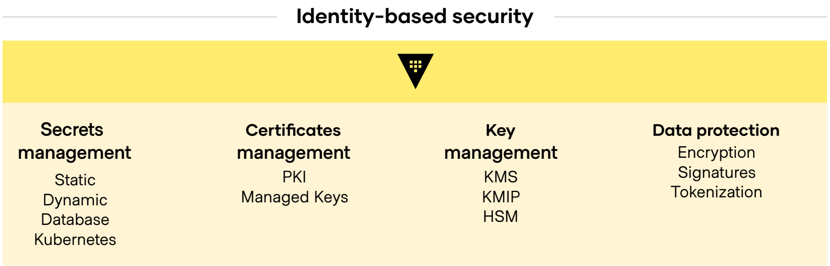
Security layer

On-premises infrastructure environments have traditionally been designed as high-trust networks. Robust firewalls secure the perimeter of the network, and traffic inside is implicitly trusted. In modern cloud environments, this traditional security perimeter no longer exists, so trust must be based on the identity of the requesting entity. IP addresses are no longer a scalable or safe form of identity since they are highly ephemeral in a cloud environment.

As a result, the security layer of a cloud operating model should be built on the principles of zero trust security, which means to trust nothing and authenticate everything. In other words, all traffic inside your network needs to be authenticated (ideally via mTLS) every time someone wants to access infrastructure resources or a service talks to another. Hence, identity becomes the cloud control point for security.

Manage secrets and protect data with HashiCorp Vault

HashiCorp Vault provides security automation to manage access to secrets and protect sensitive data. The secrets management layer aims to establish a centralized security automation platform as a shared service to the rest of the organization. This standardizes the workflow to manage the entire lifecycle of secrets management from acquisition to rotation to revocation, while auditing ensures those secrets meet your internal security compliance requirements. Built around identity-based security as the foundation of an organization's zero trust security implementation, Vault covers the core cloud security use cases organizations face:



Identity-based security

Secrets management
Static
Dynamic
Database
Kubernetes

Certificates management
PKI
Managed Keys

Key management
KMS
KMIP
HSM

Data protection
Encryption
Signatures
Tokenization

- **Secrets management:** Manage the creation, rotation, and revocation of all types of secrets across all major cloud platforms.
- **Certificate management:** Generate and manage all aspects of certificate lifecycles.
- **Key management:** Generate and manage keys according to major encryption and communications protocols.
- **Data protection:** Employ tools to help protect data in motion and at rest across the entire cloud estate.

Through all of these use cases, Vault is the identity broker that enables platform teams to securely store and tightly control access to secrets, credentials, tokens, passwords, certificates, encryption keys, and data. First, organizations build out shared services enabled by integrations with more than 100 partners and software tools to implement centralized secrets management. This is the foundation to deliver more advanced security services. Teams can integrate encryption as a service, automated secrets rotations, and advanced data protection across their entire hybrid estate. This embeds security considerations within the platform, so product teams need only “plug in” to the provided APIs to ensure their service meets corporate security standards.

At the security layer, these central shared services allow platform teams to deploy common workflows to fit their core needs:

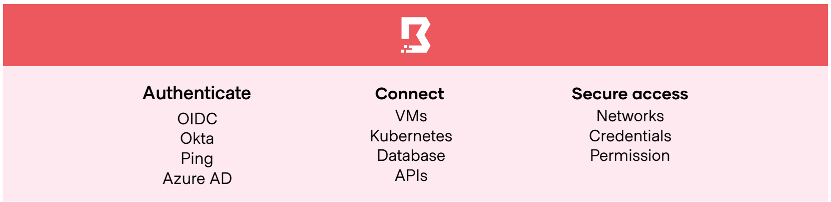
- **Unified workflow management:** Leverage workflows to centrally generate, store, access, and manage secrets and credentials, whether they are static or dynamic, along with advanced data protection and key/certificate management.
- **Reliability and scale:** Enable disaster recovery and high availability through failover to standby Vault clusters by replicating secrets and policies across any number of datacenters and cloud regions.
- **Policy and security:** Use the Sentinel policy as code framework to meet policy and compliance requirements. Enable fine-grained policy decisions, access controls, and data encryption with centralized key management to simplify encrypting data in transit and at rest across multiple clouds and datacenters.

- **Governance, risk, and compliance:** Meet governance and regulatory requirements including SOC-2 and FIPS 140-2 compliance.
- **Visibility and optimization:** As the secrets management system of record, Vault enables visibility and optimization by creating and exporting audit logs. These can be aggregated to generate insights in dashboarding and SIEM tools, enabling teams to meet additional security, reporting, and compliance requirements.
- **Integrations and APIs:** Integrate and deploy security tools with support from cloud identity and access management providers from Amazon Web Services (AWS), Microsoft Azure, and Google Cloud as well as identity providers such as Okta and Ping.

Secure and manage access with HashiCorp Boundary

Secrets management mainly handles the machine-to-machine side of zero trust security. Another solution is required for human-to-machine access. Traditional solutions for safeguarding user access, like distributing and managing SSH keys, VPN credentials, and bastion hosts either don't scale well in the cloud or have significant risks and require substantial manual effort. HashiCorp Boundary is a secure remote-access solution that solves these challenges. Boundary provides an easy way to safeguard access to applications and critical systems with fine-grained authorizations based on trusted identities. Boundary supports organizations' identity-based secure remote-access automation use cases to manage access across any environment:

Identity-based secure remote access



Authenticate	Connect	Secure access
OIDC	VMs	Networks
Okta	Kubernetes	Credentials
Ping	Database	Permission
Azure AD	APIs	

- **Authenticate** with any trusted identity provider.
- **Connect** to dynamic infrastructure and services.
- **Secure access** to any remote system, host, or service such as SSH, Linux, or databases that can integrate with Vault to create single-use, just-in-time credentials to reduce credential sprawl.

By leveraging Boundary, platform teams can connect users to clouds, local datacenters, and low-trust networks without exposing the underlying network. In a modern secure remote-access session, the user will never be directly on the network, and they'll have precise permissions so they can access only the infrastructure they need, when they need it.

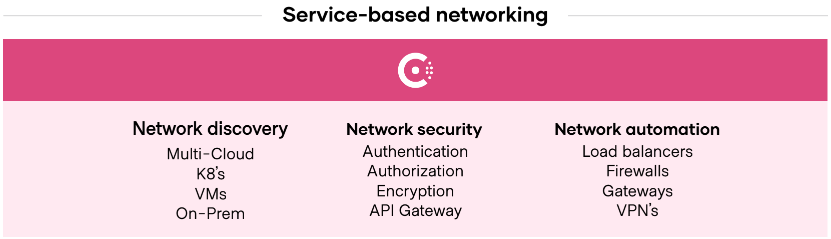
Networking layer

Networking in the cloud is one of the most difficult challenges for organizations. Engineers must navigate dynamic IP addresses, manage a significant growth in east-west traffic for microservice implementations, and adjust to the lack of a clear network perimeter. With the volume and variety of networking requirements in cloud environments, legacy ticket-driven workflows should be replaced by automated processes to unlock the speed and scaling promises of cloud-native applications.

The first step in adopting a cloud operating model at the networking layer is to find a solution that drives the discovery, registering, and connection of your services, applications, and environments. Organizations can then use this foundation to facilitate proper identity-based zero trust networking policies and advanced networking systems, such as a service mesh.

Securely connect applications with HashiCorp Consul

HashiCorp Consul enables platform teams to overcome these networking challenges by automating service networking across their entire cloud estate. Consul covers the core networking use cases that organizations must adopt as part of a cloud operating model:



Service-based networking

Network discovery	Network security	Network automation
Multi-Cloud K8's VMs On-Prem	Authentication Authorization Encryption API Gateway	Load balancers Firewalls Gateways VPN's

- **Network discovery:** Provides a service directory that enables registration, discovery, and monitoring of cloud services across all major platforms.
- **Network security:** Creates consistent encryption and authorization to secure and enable identity-based access to applications and services, as well as observability and resilience for application traffic.
- **Network automation:** Reduces the time to deploy applications and eliminates manual processes by automating complex networking tasks.

To start, Consul facilitates the discovery, registration, and connection of all services in your application environment. This registry provides a "map" of what services are running, where they are, and their current health. Next, these connections can be secured based on service identities (not IP addresses), a key component in a zero trust architecture. This enables proper authorization and access to only required services as opposed to full network access.

From there, platform teams can reduce the operational complexity of existing networking infrastructure through automation. Instead of a manual, ticket-based process, Consul automates network operations using Terraform to execute changes based on predefined tasks so product teams can independently deploy applications while platform teams can rely on Consul to handle downstream automation requirements.

As network scale expands, network complexity grows exponentially. To cope with this complexity, Consul can provide a service mesh as a central networking control plane. This service mesh contains all necessary networking services including service discovery, secured service-to-service communication, and traffic management — all bundled in one interface. This enables a networking platform that developers can use to leverage the secure network layer without needing to manually deploy or understand all of the underlying technology.

At the networking layer, Consul enables the creation of central shared services to allow platform teams to deploy common workflows to fit their core needs:

- **Unified workflow management:** Acts as a centralized registry that discovers, tracks, and monitors your services as your single source of truth.
- **Reliability and scale:** Helps teams achieve requirements for disaster recovery and high availability by providing multi-datacenter federation, automatic failover, redundancy of services across regions, and multi-tenancy to scale operations.
- **Policy and security:** Provides service identity to meet policy and compliance requirements, offering end-to-end service traffic encryption, mutual authentication of service communications, and granular application-level access control policies.
- **Governance, risk, and compliance:** Provides admin partitions for enterprises to reduce operational complexity and scale with governance policies in place.
- **Visibility and optimization:** Creates and exports audit logs and observability of service traffic data to meet additional security and compliance requirements.
- **Integrations and APIs:** Supports APM providers, integrations with Terraform and Vault for unified workflows, and cloud providers like AWS, Azure, and Google Cloud.

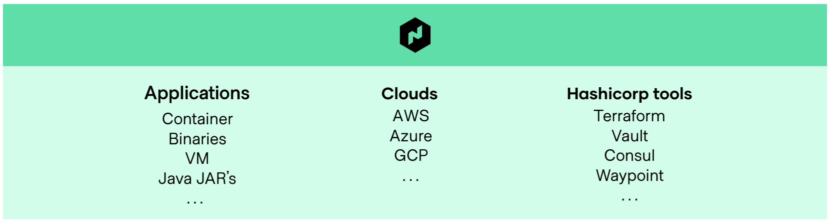
Application layer

Application teams strive to deliver code and software updates faster as organizations start building in the cloud. Traditional IT practices can hold them back, however, requiring developers to go through complex ticket-driven workflows that take days, weeks, or months to deploy new applications. On top of this, even as new cloud-native applications are built, legacy application deployments still need to be managed in parallel, ideally through one workflow and interface. New tools are required to facilitate the needs of building, deploying, and running multiple applications in the cloud.

Standardize application scheduling with HashiCorp Nomad

HashiCorp Nomad is a flexible application scheduler that can deploy and manage both traditional and modern applications for all types of workloads. It serves as a coordination layer between developers and operators, providing automated deployment (no more tickets), efficient server utilization (cost reduction), and easy workload management. It helps with the common use cases organization experience as they schedule their applications across multiple formats, operating systems, and environments:

Application scheduling as code



- **Applications:** Handles filesystem isolation, networking, and resource management platforms.
- **Clouds:** Schedules workloads across all major cloud environments.
- **HashiCorp tools:** Integrates with all HashiCorp solutions to provide uniform cloud workflows.

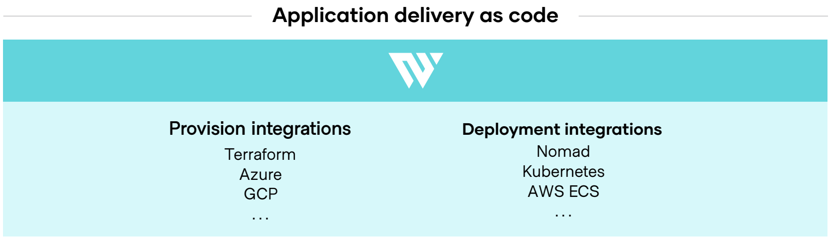
Nomad is flexible enough for all major cloud environments, operating systems, and workload types, with functionality to support new workload types in the future. Plus, Nomad's integrations with Terraform, Vault, Consul, and Waypoint enable consistent delivery of applications, while meeting necessary compliance, security, and networking requirements.

At the application layer, Nomad enables the creation of shared services that allow platform teams to deploy common workflows to fit their core scheduling:

- **Unified workflow management and integrations:** Provides methods to consolidate your application delivery workflows across clouds, operating systems, and workload types.
- **Policy and security:** Enforce checks/policies to ensure compliance with Sentinel and zero trust best practices with SSO for improved security integrations with cloud apps.
- **Visibility and optimization:** Includes dashboards to monitor performance and visualize workloads, including critical event data such as state changes or failures.
- **Reliability and scale:** Supports autoscaling to optimize how clusters use compute resources, Dynamic Application Sizing to further optimize compute resource usage, federation support for multi-region deployments ensuring high availability and uptime, and strong support for edge computing use cases.
- **Governance, risk, and compliance:** Features redundancy, automated backups, and SOC 2 compliance.

Standardize application delivery with HashiCorp Waypoint

Application delivery was once done using a mix of tools, but the process can now be streamlined intelligently through the concept of application delivery as code. HashiCorp Waypoint provides this modern workflow to build, deploy, and release applications across any runtime. Waypoint supports the common application delivery use cases organization now face in the cloud:



- **Provisioning integrations:** The top-line app delivery workflow leverages Terraform to provision infrastructure that application teams use to make their apps function across public clouds.
- **Deployment integrations:** Use templates that communicate with cloud services to standardize the release process across an organization's chosen orchestrator and workload types.

Organizations generate complexity and fragmentation as they create new systems of engagement and modernize large swaths of their application estate. Waypoint addresses this by establishing an end-to-end golden workflow for production deployments. This helps empower platform teams to deliver a single path to production that suits the needs of development teams, InfoSec professionals, and compliance audits. As a result, organizations reduce wasted effort as their application pipeline no longer requires repetitive scripting and instead promotes more frequent and higher-quality releases.

Delivering the value of a cloud operating model

Leveraging a cloud operating model helps unlock the fastest path to value in a modern multi-cloud environment. This value takes the form of not only speed in development, but also cloud and resource optimization, overcoming skills gaps, and creating a hardened security posture that reduces risk at every level of the cloud. Organizations employing a cloud operating model can expect several important outcomes:

Greater development speed:

- Shared services focused on application delivery velocity leads to shipping software ever more rapidly with minimal risk.
- Embracing DevSecOps and other agile practices helps continuously deliver increasingly ephemeral and distributed systems.

Increased efficiency:

- Reduced costs due to reused expertise resulting in consistent workflows being applied consistently in all environments.
- Leveraging a service catalog of common services for development teams to use adds new capabilities based on feedback over time and the relevant stage of the maturity model.

Reduced risk:

- Integrated policy and governance tooling match the speed of delivery with compliance to manage risk.
- Security and compliance requirements embedded within the platform itself accelerate product deployments.

A cloud operating model in practice

HashiCorp has many examples of organizations' successes at each layer of the cloud operating model:

cielo

Infrastructure: Cielo, a leading electronic payments provider in Latin America, was able to go to market 5 times faster due to provisioning infrastructure taking 90% less time. Using Terraform, Cielo's development teams were more efficient, collaborated more easily, and didn't need to reinvent the wheel every time they needed infrastructure. Ultimately, Cielo created a self-service infrastructure platform — with policy and security measures in place — that no longer requires lengthy waits for IT to approve requests.

STARBUCKS

Security: Coffee giant Starbucks needed a highly scalable way to manage secrets using an identity-driven workflow to handle the company's massive edge footprint. With Vault, it was able to scale its secrets management to more than 100,000 edge devices. The company implemented secure secrets and credentials that are short-lived and provide least-privileged access across its entire platform, reducing the impact of any compromised credential or system.

stripe

Networking: Stripe, a SaaS-based global financial services company, used Consul to increase the number of infrastructure nodes it could use to process payments by 30x. Using a common architecture with 5 server nodes per datacenter, Stripe used Consul to support end-to-end service discovery features and multi-region federation across multiple virtual domains, each with different security requirements identified by domain, region, and environment.

ROBLOX

Applications: Due to its rapid growth, video game company Roblox adopted Nomad to help improve resource management, efficiently schedule tasks, handle containers, and support high-release velocities at scale. As a result, the company doubled the efficiency of its existing bare-metal servers, saved millions of dollars in licensing costs, and was able to deploy applications and onboard new developers faster, resulting in high availability across more than 180 countries.

These are just some examples of the benefits organizations can gain by implementing a cloud operating model with HashiCorp. We invite you to learn more about how other organizations, including Vodafone, Capital One, Progressive, AthenaHealth, 3M, and others, are working with HashiCorp to bring their cloud operating model to life at www.hashicorp.com/case-studies.

Conclusion

The cloud is the most fundamental shift in computing over the last 20 years. But while it promises dramatic advances in how organizations innovate, respond to market trends, and connect with their customers and employees, it also requires significant changes in how applications are built, deployed, and managed.

Coping with those changes requires core shifts in 3 key areas to promote greater efficiency through unified workflows:

- **People:** Shift to platform engineering practices to enable cloud and platform engineers to automate with code and treat operations as a software problem.
- **Processes:** Establish centers of excellence for infrastructure, security, networking, and other functional areas for self-service delivery of capabilities.
- **Tools:** Switch to dynamic environments that use tools that support the increasing ephemerality and distribution of infrastructure and applications to power critical workflows rather than being tied to specific technologies.

This may seem daunting, which is why successful organizations create platform teams and apply a maturity model to manage and benchmark this process.

Platform teams create a center of excellence that supports cloud adoption by delivering shared services across each layer of the cloud. By employing a maturity model, these teams can establish a detailed roadmap laying out where your organization currently stands and what needs to be done to reach its goals. This roadmap helps teams throughout the organization make better decisions faster. Just as important, it enables benchmarking progress along the way. Without a thoughtful roadmap and clearly articulated process, cloud adoption can be significantly slower and more expensive.

Adopting a cloud operating model is a critical step for enterprises aiming to maximize their digital transformation efforts. The cloud strategy/program office is evolving away from ITIL-based control points — focused on cost — toward becoming self-service enablers focused on speed. They enable product teams to deliver new business and customer value efficiently, at speed, and with reduced risk.

The HashiCorp suite of products provides solutions for each layer of the cloud infrastructure to enable platform teams to successfully lead this shift to a cloud operating model. For more than a decade, HashiCorp has partnered with thousands of organizations around the world to make their cloud operating model a reality. Our portfolio of open source and commercial products have been downloaded more than 450 million times in the last year alone, and our ecosystem includes hundreds of partners and more than 3,000 integrations.

Learn more about how HashiCorp can make the cloud operating model a reality for you at www.hashicorp.com.

HashiCorp

www.hashicorp.com