# A Leadership Guide to Multi-Cloud Success for Federal Agencies

June 2022

HashiCorp

# Table of Contents

# Executive Summary

1

# Executive Summary

Organizations in the public sector are moving from a single-cloud community to a multi-cloud one. Examples of this have already been seen in the intelligence community and Department of Defense space with the introduction of the Commercial Cloud Enterprise (C2E) and Joint Warfighter Cloud Capability (JWCC) contracts. These are just two examples of the massive shift that is occurring across the entire public sector as it modernizes to integrate multiple cloud providers. Leadership needs to plan for the impact this will have on the way programs are staffed and the way systems are engineered and tested. Successful multi-cloud adoption requires thoughtful, purposeful action. A broader availability of cloud service providers (CSPs) means that federal organizations must become smarter and more discerning cloud customers.

Multi-cloud brings several significant challenges. First among them is skills shortages. The breadth and depth of multi-cloud expertise requires comprehensive training, which takes time and sustained investment. As a result, industry and government have spent the last several years building the public sector's current cloud-capable workforce with skills designed for working with only a single CSP. Skills shortages are followed by difficulty in governance, inconsistent results across the organization, and continued issues with cost optimization.

The challenges are not insurmountable. Highly effective approaches include moving the cloud architecture control points up and out of any one CSP, shifting to platform teams and a platform mindset, equipping these teams with best-in-class multi-cloud tools, and taking a pragmatic approach to portability across clouds. HashiCorp and its cloud infrastructure automation tooling were created for just this purpose, and we are committed to supporting federal agencies on their multi-cloud journey.

For more information, please contact:

Chris Carroccio
Federal Civilian Sector
HashiCorp
hashicorpfederal@hashicorp.com
https://www.hashicorp.com/industries/public-sector

# Introduction

2

# Introduction

HashiCorp is a leader in multi-cloud infrastructure automation software. The HashiCorp software suite enables organizations to adopt consistent workflows and a standardized approach to automating the critical process involved in delivering applications in the cloud: infrastructure provisioning, security, networking, and application deployment. HashiCorp's open source tools Vagrant™, Packer™, Terraform®, Vault™, Consul®, Nomad™, Boundary, and Waypoint™ were downloaded approximately 100 million times during the past fiscal year. Enterprise and managed service versions of these products enhance the open source tools with features that promote collaboration, operations, governance, and multi-datacenter functionality. The company is headquartered in San Francisco, though 90%of HashiCorp employees work remotely, strategically distributed around the globe.

HashiCorp solutions complement services from the CSPs. This gives us unique insights into private industry's trials and successes with multi-cloud. Federal organizations have an opportunity to go into this multi-cloud world with their eyes open, and apply proven best practices.

Many public sector organizations have operated a homogeneous cloud environment, often leveraging a single CSP exclusively. With multi-cloud now not only an option, but in many cases an imperative, there is an opportunity to take a broader view of CSPs and consider the advantages for any available services as well as the best way to implement them.

Multi-cloud also requires a workforce evolution. The choices and nuances of multi-cloud mean this evolution needs to happen in the developer base, program oversight, and program leadership. Managers will need to lead through intentional acquisition language, thoughtful systems-engineer guidance, purposeful testing, and oversight.

Unlike companies in private industry, public sector organizations are not monolithic entities that can put out simple guidance and quickly orient the entire workforce to these new expectations. Instead, they are often made up of disparate and complex workforces. This reality requires methodical approaches to key provisioning, security, and networking workflows as well as an emphasis on shared services that make outcomes such as shared automation accessible to the full spectrum of users.

HashiCorp is committed to being a strategic partner of the federal government. We bring lessons learned from our most sophisticated customers and help you implement our industry-leading products to accelerate your multi-cloud journey.

This whitepaper lays out the larger industry trends around multi-cloud, the close alignment and instrumental role HashiCorp solutions play in successful multi-cloud organizations, the recommendations for achieving integration, interoperability, and portability across cloud service providers and translates these relevant best practices for multi-cloud to federal agencies.

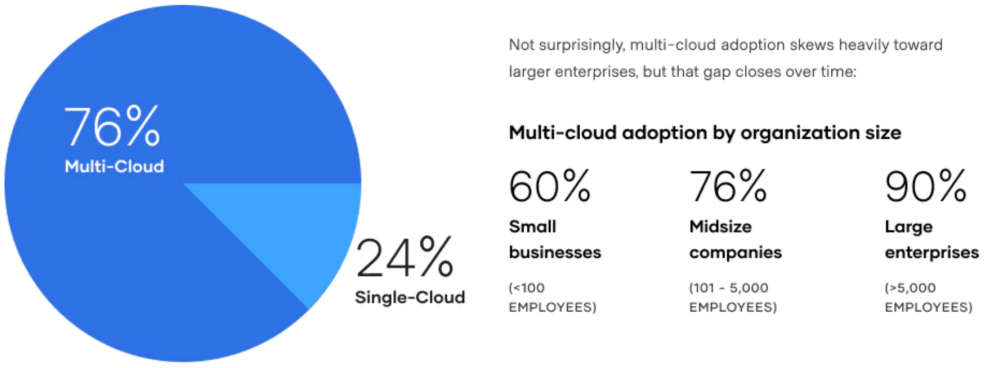
# The Multi-Cloud Era is Here

3

# The Multi-Cloud Era is Here

## Multi-Cloud Is the Reality Today

The use of multiple CSPs is inevitable over time, due to both organic and inorganic factors. According to *HashiCorp's 2021 State of Cloud Strategy Survey*, 76% of survey respondents are already using more than one cloud. What's more, the larger the organization, the more likely they are to be using multiple clouds.




*Source: HashiCorp State of Cloud Strategy Survey*

In the private sector, multiple clouds are the current reality — or are an inevitable end-state for various reasons. One example is organic adoption, where different development teams have experimented with various providers and now have apps running in production across multiple CSPs. This usage often occurs outside the purview of central IT. Another is a merger or acquisition, where a company's cloud strategy and roadmap suddenly churns when they are combined with another firm. In some cases organizations choose to pursue diversification, embracing the pattern of a "primary" cloud for a certain class of workloads, while a "secondary" cloud is used for other scenarios.

In the federal government space, a combination of changing cloud needs and mandates to leverage multiple CSP is driving much of this transformation  For the IC, the move from the C2S acquisition to the C2E acquisition is the driving factor. The availability of multiple FedRAMP CSPs has already enabled organizations in the public sector to begin their multi-cloud journey.

# Multi-Cloud Challenges

## Different Cloud Service Providers Have Different APIs

Each CSP has its own set of APIs that correspond to its respective services. Foundational capabilities — shown below — have different implementations and require users to learn and master the subtle differences in implementation for familiar concepts.

| Delivery Layer | Private Cloud | aws | | | | IBM |
|---|---|---|---|---|---|---|
| **Run** Development → | vSphere | EKS / ECS Lambda | AKS / ACS Functions | GKE Cloud Run | OKE Functions | K8s Service Code Engine |
| **Connect** Networking → | Hardware | Cloud Map | Proprietary | Istio | Service Mesh | Istio |
| **Secure** Security → | AD/LDAP | AWS IAM | Azure AD | GCP IAM | OCI IAM | Cloud IAM |
| **Provision** Operations → | Terraform | Cloud Formation | Resource Manager | Cloud Dep. Manager | Resource Manager | Cloud Resource Manager |

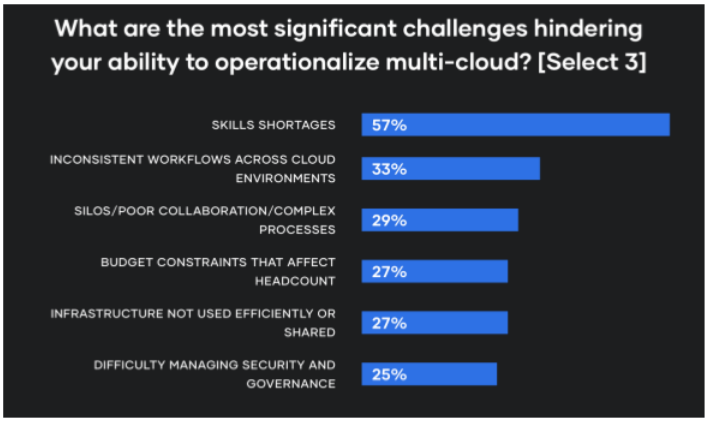
Even services that are quite similar across cloud providers, such as a secrets management service or a Kubernetes runtime, can vary wildly in important ways. The further you move up the stack — into advanced databases and serverless computing — the more divergent the APIs become. These technical differences drive human-centered problems for multi-cloud organizations.

## Subtle Differences Drive Complex Organizational Challenges

Multi-cloud organizations face a common set of challenges. According to the *HashiCorp 2021 State of Cloud Strategy Survey*, the most significant challenge is skills shortages, followed by organizational process and consistency challenges. Of these, four challenges are most important for federal organizations in this new multi-cloud world and warrant a deeper look.

# Top Challenges to Operationalize Multi-cloud

When it comes to the challenges hindering operationalizing multi-cloud, respondents had similar concerns.




*Source: HashiCorp State of Cloud Strategy Survey*

## Skills Shortages

CSPs offer hundreds of services; the multi-cloud ecosystem is even larger, and requires deep proficiency across scores of products, processes, and technologies. Each CSP is just different enough that the skills and expertise of technical staff may not translate one-to-one across clouds. As a result, it is difficult to find talent proficient in more than one cloud. As organizations use multiple CSPs, it becomes exponentially more difficult to find engineering talent, standardize “golden workflows”, and achieve desired outcomes.

## Cultural Transformation Is Siloed and Uneven

Purely organic usage of CSP services becomes, over time, untethered from a common culture, or a common way of working. This is especially true of large, distributed teams. Pockets of exemplary behavior exist, but they are not uniform. For government organizations where the teams are contractors from different companies, these top-performing groups exist in a sea of “technically acceptable” teams. The bottom 90% of teams can’t be managed the same way as the top 10%. This results in applications and architectures that are more anti-pattern than best practice. These silos make it almost impossible to achieve consistent outcomes, at scale, across application teams.

## Governance Becomes Difficult to Manage

As individual teams adopt isolated infrastructure as code and automation approaches, the organization can become blind to poor-quality automation. For government organizations with federated structures, automation governance can quickly become impossible. Fragmented automation oversight is effectively no oversight, allowing poorly done automation to quickly inflame a problem. In fact, these issues may spread to other teams that lack the skills to recognize they are perpetuating a vulnerability or anti-pattern. Making the situation even difficult, the skills needed for oversight are in just as short of supply as the skills needed to perform cloud automation.

## Cost Is not Optimized

Development and test environments often run idle, and are rarely de-provisioned after the fact. Some infrastructure is “over-provisioned” (i.e. an XL instance is provisioned when a medium will do). Over time — and at scale — millions of dollars are wasted in bloated cloud budgets.

# Achieving Multi-Cloud Success

4

# Achieving Multi-Cloud Success

Given the challenges of multi-cloud, it is logical for an IT leader to say, “I need this status quo to change, and I need to avoid past mistakes as we bring new CSPs online.”

Large organizations are focusing on four key practices to re-position themselves for multi-cloud success:

1. **Move the new architectural control points up and out of any one CSP.** Each layer of the stack (infrastructure, security, networking, and applications) is built around a new pattern, a different control point. The implementation of new control points is unique to a CSP. Focus on workflows that elevate these control points outside of a specific CSP service.
2. **Shift to platform teams and a platform mindset.** Empower a small group with the right tools to set up the entire organization for multi-cloud success.
3. **Equip enterprise teams with best-in-class multi-cloud tools.** Partner with ISVs that excel at multi-cloud deployments. In addition to optimized tools, ISVs can bring a wealth of engineering talent. Best-in-class ISVs with self-managed and cloud-managed offerings are key to achieving better outcomes across multiple clouds.
4. **Be pragmatic about vendor lock-in.** Ensure the business case for portability is appropriate and understand the costs of moving a workload from one CSP to another.

## Elevate Control Points Out of any One CSP

Static infrastructure was designed for relatively infrequent updates to production and constant user traffic. In contrast, dynamic infrastructure — built atop APIs with cheap utility pricing — enables rapid horizontal scale and greater resiliency. Cloud-native applications take advantage of these characteristics with different architecture patterns. Each layer of the stack features a different architectural control point:

- Applications - Containers / Continuous Delivery
- Networking - Service Registry
- Security - Identity
- Infrastructure - Instructure as a Code

Unsurprisingly, each CSP offers proprietary services for these control points.

In the multi-cloud era, while the infrastructure is not static, the consumption experience has become stable for single cloud users. Each major CSP has created an optimized version of its API-driven infrastructure. Different CSPs have different APIs for core infrastructure services, and wildly different APIs for highly differentiated services. These services might be the most infrastructure-efficient version of these capabilities, but they are tightly tied to that CSP and are incompatible with multi-cloud scenarios.

Organizations that want to be able to operate on multiple clouds are evolving and moving the definition of dynamic up the stack. In the multi-cloud era, the control points have moved from "static to dynamic" to "dynamic on dynamic." To achieve flexibility across CSPs, organizations are moving the control points up and out of any one CSP. This is a delicate balance, since it means consciously sacrificing some of the infrastructure optimization of a single cloud and knowingly accepting the need for new skill sets and a new mix of staffing.

## Shift to a Platform Teams and a Platform Mindset

### The Platform Team

Platform teams are a best practice to deliver outcomes at scale across a range of infrastructure APIs. These teams are at the core of how multi-cloud services are managed successfully. A platform team is a highly specialized team of engineers who focus on standardizing the organization's infrastructure APIs. The team is composed of platform engineers and a platform product manager. Their collective role is to create a common set of infrastructure and service APIs for all other enterprise application development teams to use. This model abstracts the increasing complexity of the multi-cloud architecture away from other development teams. Platform teams are also responsible for site reliability engineering (SRE), making sure the platform meets organization uptime, resiliency, reliability, and security goals.

Platform engineers require a combination of infrastructure and software engineering skills. Because SRE treats operations as a software problem, platform engineers are coders. The addition of new CSPs adds organizational complexity. The platform engineering team should be staffed up accordingly. Practitioners with expertise on a CSP are valuable; engineers skilled

across clouds are quite rare. It is easier to staff a team of AWS engineers and a second team of Microsoft Azure engineers than it is to staff a full team of engineers who are experts on both.

The primary advantage of a platform team is it concentrates hard-to-find cloud expertise. This team, leveraging best-in-class multi-cloud enabling technology, can be a significant force multiplier for the organization. Instead of fragmenting limited talent out to a few programs for isolated gains, platform teams can help organizations methodically deliver multi-cloud outcomes at scale.

## The Platform Product Approach

Organizations should run their platform as a product. That idea revolves around the key principle of user centered design. When building a product, seek to understand the needs of users. Don't build something and then expect internal development teams to conform to these ideas. Platform teams should focus on pragmatic approaches to key areas:

1. **Define and measure reliability, including SLIs, SLOs, error budgets:** It is vital for developer teams to trust the platform team, and trust is built over time. A reliable and predictable set of services gives developers confidence that the platform will be available and able to keep their applications running at scale.
2. **Continuously improve, reducing toil and increasing automation:** Seek to automate as much as possible. Automation results in consistency, which leads to efficiencies as more developers push applications to the platform. Each automated workflow reduces the labor hours needed to achieve a new outcome later.
3. **Provide self-service portal/API for users:** Big organizations tend to build and manage highly functional platforms that handle networking, security, infrastructure, and much more. This allows development teams to simply bring their applications to the platform and wire them up in a standard way.
4. **“Shift left” to reduce friction from InfoSec and compliance:** Security, compliance, encryption, auditing, and other InfoSec concerns should be addressed deep within the platform. This way, common security and compliance requirements are met at the platform level, usually via thoughtful implementation of automation and reference architectures. Individual teams should focus their security and compliance efforts for their particular service, rather than the full stack. The end result: each application or custom service running on the platform can more easily meet the rigorous standards of the larger organization.

5. **Provide a delightful developer experience:** Developers will follow the path of least resistance. Give them useful APIs, with great documentation, and they are more likely to adopt the technologies you want them to. In complex organizations, relief from administrative hurdles is a powerful draw for developers to use approved platform services.
6. **Advocate — reach out to developers and drive awareness:** It's not enough to simply mandate the use of a given platform or technology. The platform teams need to advocate and promote the use of the platform for the right workloads and use cases.

Because a platform team is a service provider, the organization's platform approach should be aligned with business outcomes. Its success or failure needs to be measured against outcome-oriented metrics. Appropriate metrics include time to ATO, platform stability and reliability, security-workflow efficiency, and cloud-cost optimization.

## Plan for Scale in the Platform Approach

Platform teams are not uncommon in government. But too often, these teams do not plan for enterprise scale. This shows up in a narrow mindset that doesn't think big enough and doesn't address the least skilled users in the organization.

Enterprise platform teams often start with too small of a vision for the minimum viable product (MVP). They pick a few leading open source solutions and push out basic functionality. This approach can work initially. As more developers adopt the services, however, cracks can begin to form. Many open source implementations include the core functionality but lack enterprise-grade user management, security, or governance for key workflows. As these issues start to show up, platform teams must spend valuable time and effort to make them perform at enterprise scale — at significant opportunity cost. Instead of focusing on improving workflow outcomes, the platform team is investing hours on undifferentiated tasks that merely keep the lights on.

Even planned upgrades to enterprise versions at a later date can be disruptive. Instead of attempting a risky "hot swap" to better tools after reaching an MVP milestone, plan for enterprise scale from the beginning. Of course, the platform doesn't need the capacity to handle thousands of apps on Day 1, but it should be designed, built, and operated with that scale in mind. Often this means starting with a very small deployment of an enterprise-grade solution.

Contracted platform teams often fail to understand the lowest skill levels in the user base. If the acquisition is not outcome-based, teams focus on deploying the next tool rather than user adoption. Not every developer is a command-line wizard, and not everyone in oversight can read automation scripts. For the organization as a whole to succeed, platform teams need empathy for the average developer, since they make up a majority of the users, especially in government. Platform teams need to focus on making solutions accessible, abstracting complexity away from development teams, and supplying excellent documentation.

If infrastructure and workflow automation is left to individual teams, the multi-cloud initiative may fall short of success because of the skills shortage. Incorrect automation or automation of bad practice can be wildly detrimental to an organization. Automation must be configuration controlled and governed. It must be consistent and regularly updated and validated. Ad hoc automation approaches can easily spiral out of control and create mission-critical issues.

## Equip Enterprise Teams with Best-in-Class Multi-Cloud Tools

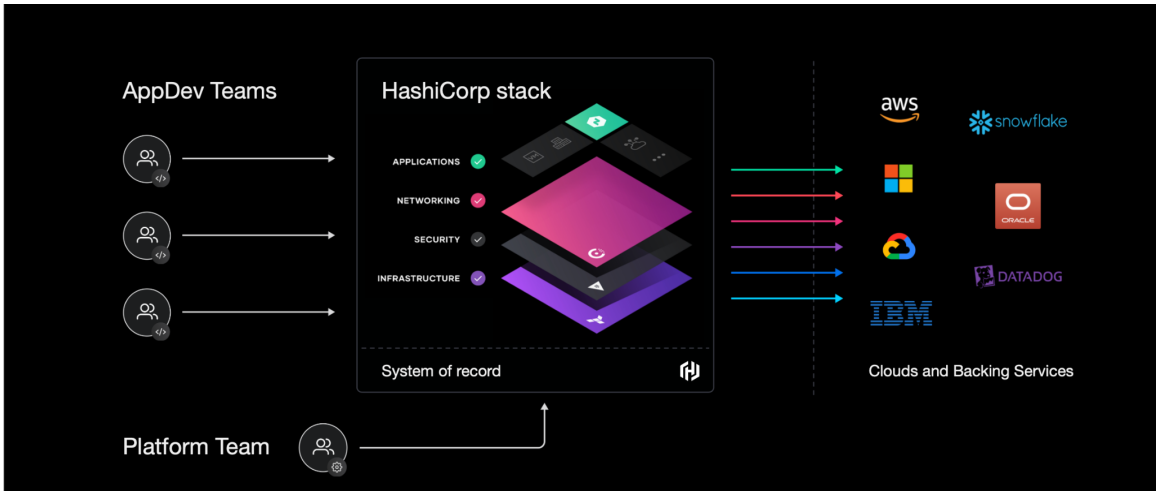
To deliver the platform approach efficiently, platform teams must be equipped with best-in-class multi-cloud tools that enable workflow effectiveness at scale. What's more, leading ISVs should be seen as part of the extended team and not just tool suppliers. Engineers from trusted vendors are the most talented people in the industry when it comes to using CSPs efficiently.

We see this pattern play out in our most advanced customers. The HashiCorp portfolio is designed to enable these platform teams to provide a cohesive, multi-cloud experience with HashiCorp Terraform, Vault, and Consul. (A deeper description of the HashiCorp stack is provided in the Appendix.)

Beyond the HashiCorp portfolio, we recommend organizations seek out market leaders in key multi-cloud capability areas. We recommend considering vendors in several categories:

- **Application delivery:** Azure DevOps, GitHub
- **Release orchestration:** CloudBees, GitOps,  Azure DevOps, Helm, Jenkins, Bamboo
- **Data services:** Confluent, MongoDB, Snowflake, and Databricks
- **Observability:** Splunk, Elastic, Grafana, and DataDog

Using these tools and collaborating with the engineers behind them is a powerful force multiplier for platform teams.




### The Benefits of the Well-Equipped Platform Team Approach

Following the platform team pattern addresses the major challenge of multi-cloud: the staff skills shortage. A platform team concentrates available skills in a small team to create enabling workflows for the rest of the organization. Further, platform teams, equipped with best-in-class multi-cloud tools, allow an organization to use fewer staff to reach its multi-cloud objectives. Additionally, standardizing key multi-cloud workflows and managing them centrally optimizes many of the governance challenges of releasing new code to production.

## Be Pragmatic about Vendor Lock

Vendor lock is a particularly tricky subject in government. Completely avoiding the risk of lock-in is impossible. Since almost all capabilities are built by contractors or come from commercial suppliers, the reality is there is always some form of vendor lock. It takes many forms: a proprietary interface in software or hardware, or a services team that has customized an open source implementation into a “snowflake” toolchain that can be maintained only by the engineers who did the original work. Even “free” software still locked in at the services-team level, can carry a high total cost of ownership, and is likely to have a high switching cost.

Multi-cloud is forcing government program managers to at least consider aspects of CSP lock-in. The guidance for many federal agencies is to  an expectation for any CSP to support

integration, interoperability, and portability. This is a good starting point, but how this manifests is not a responsibility of the CSPs. It is a responsibility of the consuming programs. Best practice is to be pragmatic and take a business case-based approach.

## Integration and Interoperability Among CSPs

Integration and interoperability are key goals not just for new CSP contracts, but also for the missions operating on them. Information sharing is vital to the success of the mission. To maximize integration and interoperability, think “API first.”

Composable architecture is a must-follow design pattern for multi-cloud. In general, integration and interoperability can be done pragmatically across missions by using “composable architectures.” Composable architectures describe a style of application architecture, one that focuses on API-driven contracts between systems and components. The emphasis on APIs leads to architectural flexibility as the bigger enterprise IT environment evolves and becomes more diverse.

Many applications still run on-premises in virtual machines and need to be exposed through APIs. The rise of SaaS products means some capabilities can be consumed on-demand, just as the CSPs are. There is an explosion of new and emerging data services. Doing composability right means when developers build new features, whatever service they need is an API call away. Platform teams need to ensure latency requirements are met at every connection point.

This API-based model requires new approaches to service management and security. Multi-cloud automation tools like those from HashiCorp can help. For example, HashiCorp Consul simplifies how services connect across clouds and HashiCorp Vault manages secrets and encryption across clouds and cloud services.

## Portability across CSPs

Application portability is hard, and it should be a non-goal in most cases. Custom code can conform to individual CSP APIs to varying degrees. Code written for higher-level services on Azure won’t easily work on AWS, for example. Even with common services based on open source projects (like Kubernetes and Postgres) the APIs used by the various CSPs are just “different enough” to make portability cost-prohibitive in most scenarios. That’s not necessarily a

problem, as long as the enterprise is aware of the issue. Portability is usually most cost effective when applied to applications that do not change often, or are not mission-critical.

When custom code is deeply tied to the CSP's APIs, the cost and effort to move that application to another CSP must be supported by a compelling business case. Often, that happens when the business has a strong desire to avoid prolonged downtime in the event of a multi-region failure by a single cloud. For example, some retail providers run their core e-commerce services on one "primary" cloud, with automated failover to a "secondary" cloud in the event of an outage. This style of multi-cloud architecture and redundancy is costly and challenging. But some retailers have deemed the extra complexity worth it to ensure business continuity —and peace of mind. For government missions, portability may be required to meet resiliency requirements.

If portability *is* required, managers need to understand that the choice of CSP services cannot be left to chance. They must closely monitor their implementation teams to ensure they are making appropriate trade-offs when selecting which APIs to use. Platform teams can lower the cost of portability by using shared APIs wherever possible. Vault and Consul for example, cut portability costs because their APIs transcend any given cloud.

# The Way Ahead

5

# The Way Ahead

Organizations in the federal government that are new to multi-cloud have an opportunity to get started on this journey the right way. There is a proven collection of best practices to help you plan and account for the coming challenges. HashiCorp and its cloud infrastructure automation tooling were created to help complex organizations successfully deliver business outcomes at scale across clouds. We are committed to being a close partner of the federal agencies, bringing lessons from our most sophisticated customers and evolving product features together with them.

We already work closely with many of the CSPs and system integrators (SIs) across all areas of the public sector. The open source versions of our tools are already the foundation of many of the cloud automation efforts going on across the community.

We are working hard to support and accelerate the multi-cloud transformation. We collaborate closely with enterprise platform teams, sharing best practices and helping these teams to scale. We are fierce advocates for platform teams as we engage with SIs and help educate the enterprise about pragmatic approaches to portability across CSPs.

# Appendix: HashiCorp Product Snapshot

6

# Appendix: HashiCorp Product Snapshot

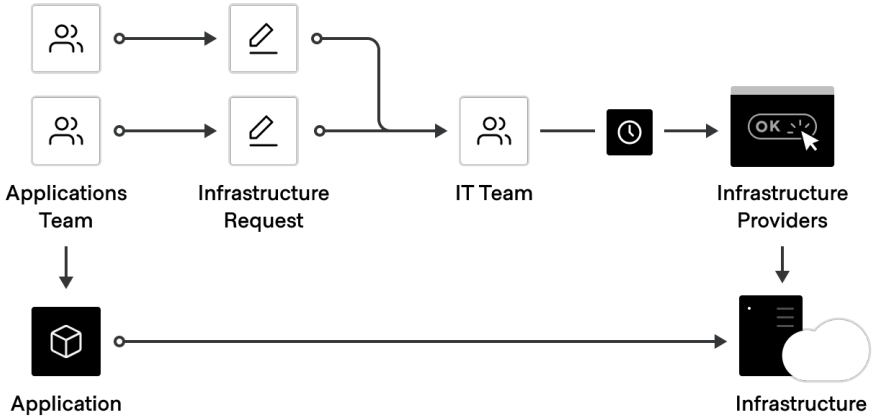
HashiCorp and its cloud infrastructure automation tooling were created to help complex organizations successfully deliver business outcomes at scale across clouds. The HashiCorp portfolio is designed to enable platform teams to provide a cohesive multi-cloud experience for a much larger organization of development teams building custom code.

## Infrastructure Provisioning with Terraform

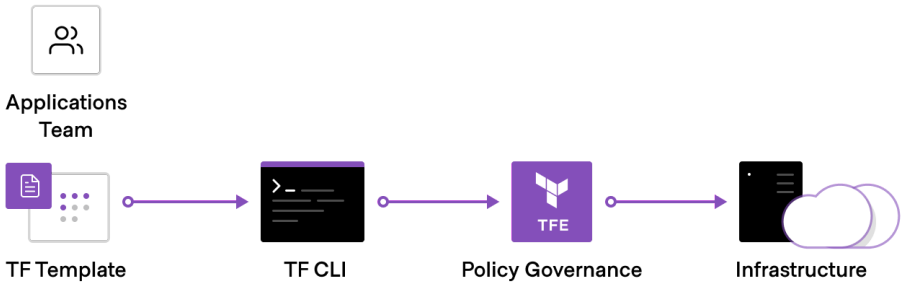
Centralized teams codify policies enforcing security, compliance, and operational best practices across all cloud provisioning. Automated enforcement of policies ensures changes are in compliance without creating a manual review bottleneck.

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.

### Before Terraform



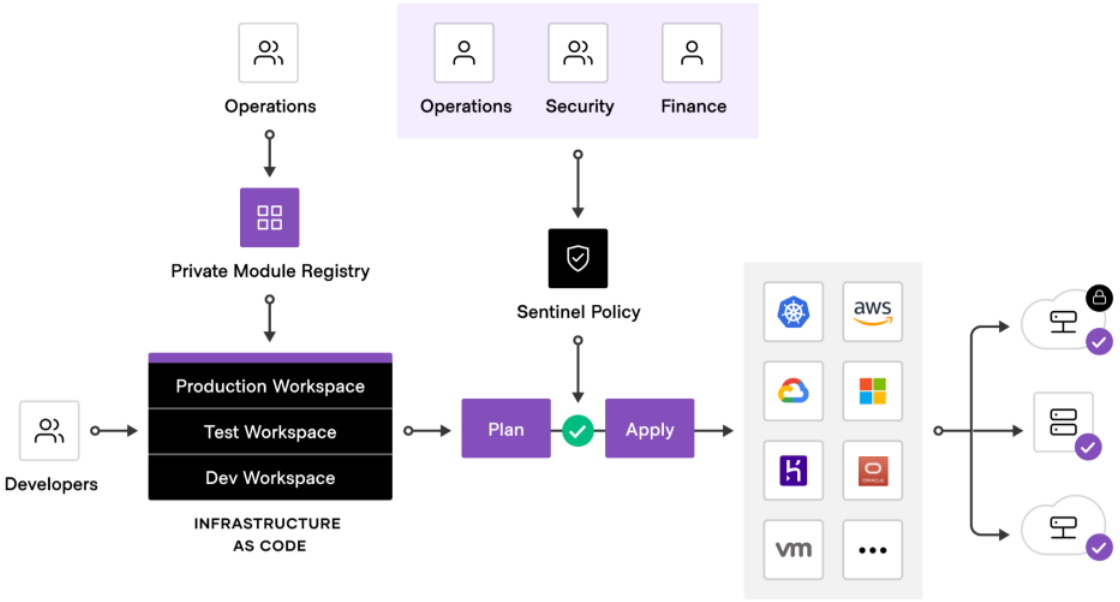

### After Terraform




To achieve shared services for infrastructure provisioning, platform teams should start by implementing reproducible infrastructure as code practices, and then layering compliance and governance workflows to ensure appropriate controls. This foundational infrastructure as code use case accelerates provisioning, reduces costs (people/time/number of steps) associated with provisioning, and improves cost forecasting and controls.

Most platform 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 because Sentinel policy is also defined as code, it enables collaboration and comprehension for DevSecOps.

The most mature Terraform deployments position platform teams as a self-service enabler of speed versus a ticket-based gatekeeper of control. At the same time, Terraform helps these platform teams ensure compliance and governance objectives are met.

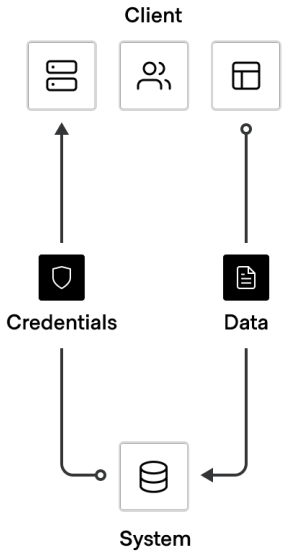



## Security with Vault

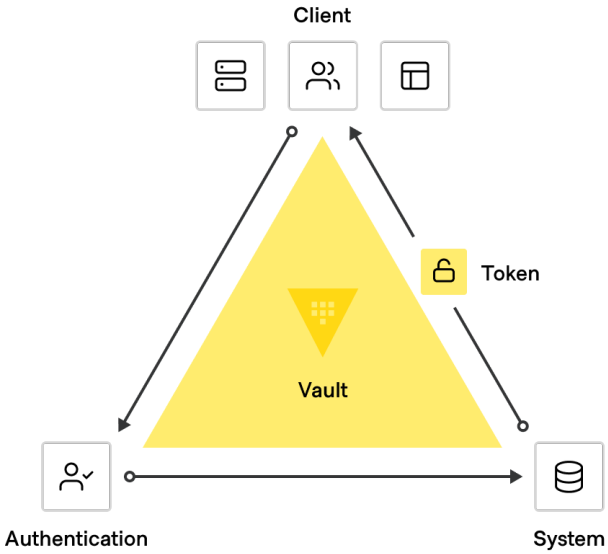
In the traditional security world, we assumed high trust internal networks, which resulted in a hard shell and a soft interior. With the modern “zero trust” approach, we work to harden the inside as well. This requires that users and applications be explicitly authenticated, then authorized to fetch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

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

Before Vault




After Vault




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

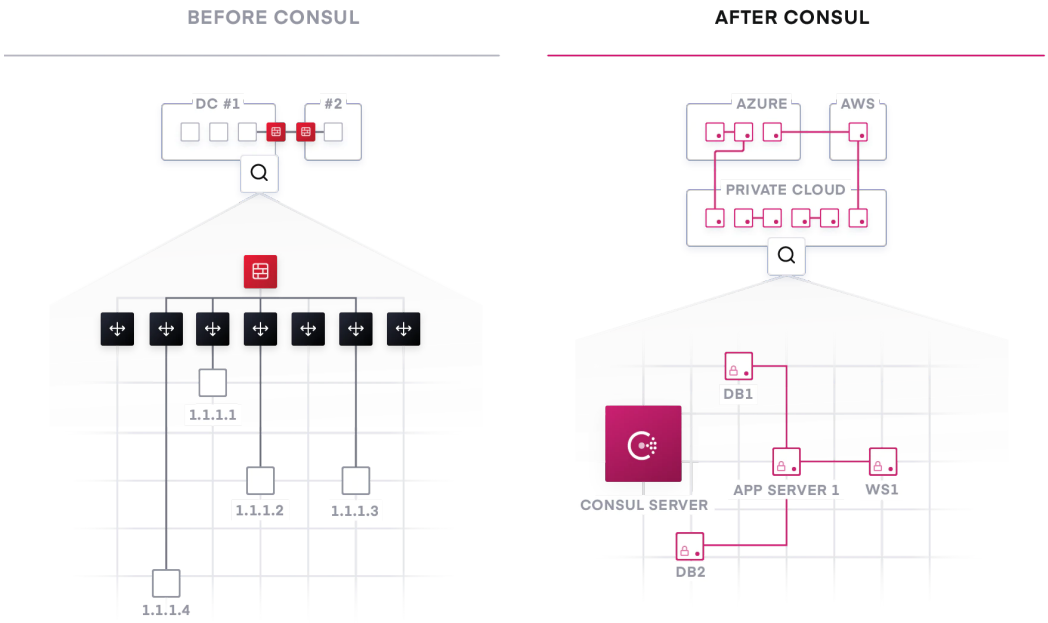
Platform teams should enable centralized secrets management services, such that every development team can simply plug into standard APIs. From there, developers can then deliver more sophisticated encryption-as-a-service use cases such as certificate and key rotations and encryption of data in transit and at rest.

## Service Networking with Consul

The challenges of networking in the cloud are often one of the most difficult aspects of enterprise cloud adoptions. In addition to being multi-cloud, organizations are also multi-runtime, and the combination of dynamic IP addresses, significant growth in east-west

traffic with the adoption of microservices, and the lack of a clear network perimeter pose a formidable challenge.

HashiCorp Consul provides a multi-cloud service networking layer to connect and secure services. Consul creates a unified networking control plane across all the abstractions used in a large organization (virtual machines, various container orchestrators, serverless engines) as well as all the infrastructure targets (private cloud, public cloud, edge).

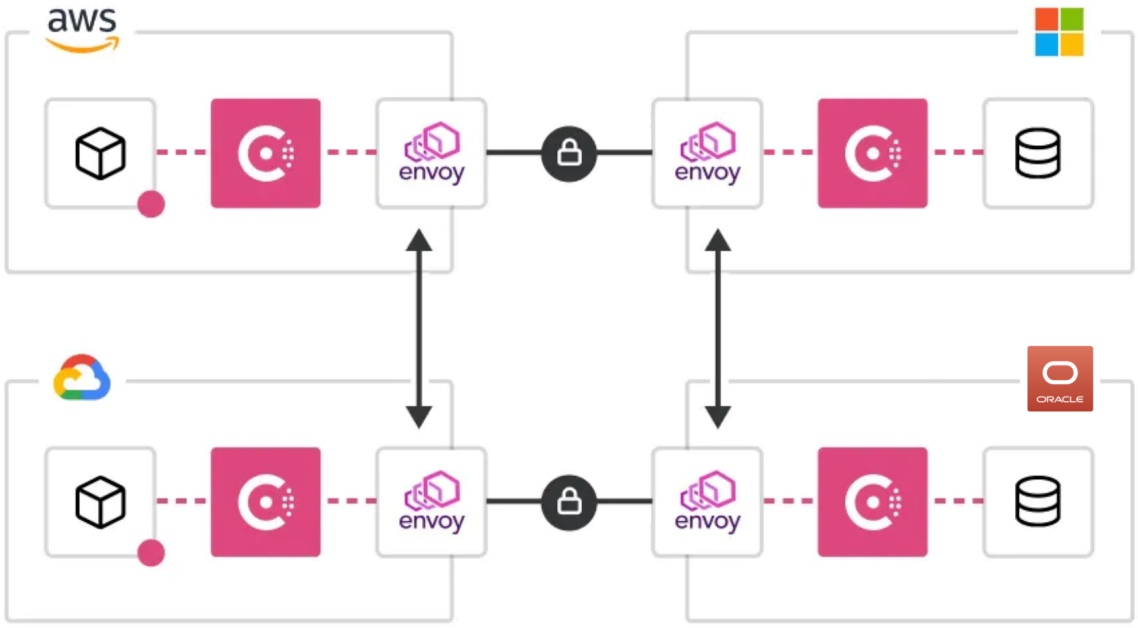


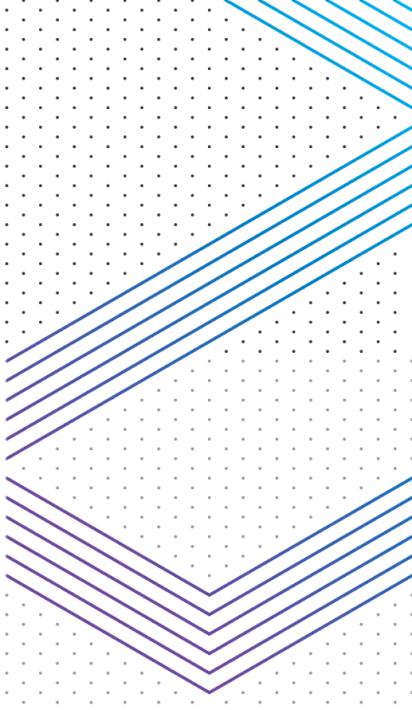

Networking services should be provided centrally, with platform teams providing service registry and service discovery capabilities. Having a common registry provides a “map” of what services are running, where they are, and their current health status. The registry can be queried programmatically to 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 patterns allow the network topology to be simplified, especially for multi-cloud and multi-datacenter topologies. Consul offers consistency across multiple clouds and platforms.

This consistent dataplane allows developers to connect their services between heterogeneous environments regardless of whether they are running on VMs in an on-premises datacenter or on a managed Kubernetes service like  Amazon EKS, Microsoft Azure Kubernetes Service (AKS), or

Google Kubernetes Engine (GKE), IBM Cloud Kubernetes Service, or Oracle Container Engine for Kubernetes.

Furthermore, Consul supports true multi-tenancy with Administrative Partitions. Multiple deployments can remain under a single control plane, allowing for consistent management and governance while maintaining autonomy and isolation for different tenants while increasing the velocity for safely/securely connecting to healthy services.





USA Headquarters

101 Second St., Suite 700, San Francisco, CA, 94105
www.hashicorp.com