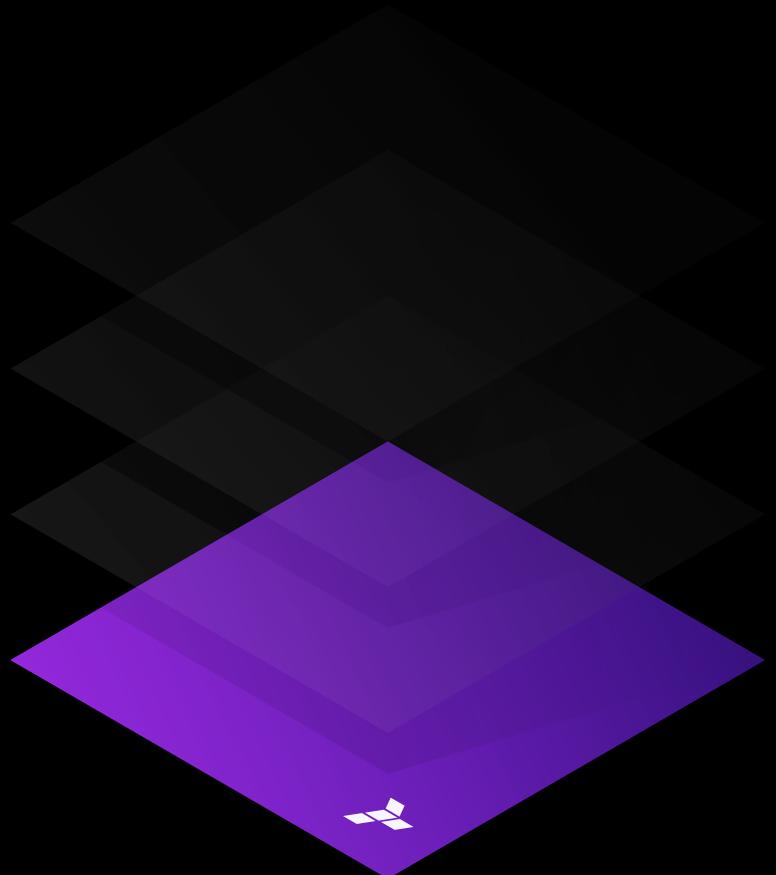




Unlocking the Cloud Operating Model: Minimizing Cloud Waste

Effectively manage cloud spend with
compliance and management



Contents

Executive Summary	1
Cloud Operating Model Overview	2
Implications of the Cloud Operating Model	3
Transitioning to a Multi-Cloud Datacenter	4
Cloud Operating Model: Minimizing Cloud Waste	6
HashiCorp Terraform	9
Conclusion	15

Executive Summary

In this white paper, we look at the implications of the [cloud operating model](#), and describe its benefits for minimizing cloud waste and effectively managing cloud spend.

This is part of a series of white papers on infrastructure as code (IaC) for [provisioning](#) and reuse, and then using IaC for standardization for [compliance and management](#).

As organizations move to the cloud, we found that of the more than 3,200 people we surveyed in our [2021 HashiCorp State of Cloud Strategy Survey](#), 39% of respondents said their organization overspent their cloud budgets. Experts have cited anywhere from [20%-35%](#) of many organizations' cloud spend is wasted. This is a huge problem as cloud efficiency and cost are top of mind for every person who looks at their cloud provider bills.

This white paper will explore the causes of cloud waste and solutions to minimize it. But first we need to understand the cloud operating model and the cloud drivers that have led to these inefficiencies.

Cloud Operating Model Overview

To thrive in an era of multi-cloud environments driven by digital transformation, enterprise IT must evolve from gatekeeping based on Information Technology Infrastructure Library (ITIL) best practices to enabling shared self-service processes for DevOps excellence.

For most enterprises, the goals of digital transformation focus on delivering new business and customer value more quickly and at a very large scale. The cloud is an inevitable part of this transformation as it presents the opportunity to rapidly deploy on-demand services with limitless scale and unparalleled compute capabilities to ultimately deliver next-generation experiences to customers.

In the cloud, however, enterprises have the challenge of maintaining their existing resources, private clouds, and datacenters while simultaneously developing new applications and services that leverage the public cloud's benefits. Many enterprises soon discover each cloud provider operates differently and they must choose among a vast array of cloud-based services.

To unlock the fastest path to value in the cloud, enterprises must figure out how to accelerate the application delivery process across every layer necessary to run an application: infrastructure, security, networking, and run-time. That makes it crucial to embrace the cloud operating model and fine-tune people, processes, and tools to that model.

Implications of the Cloud Operating Model

An essential implication of the transition to cloud is implementing a new operating model to accommodate the shift from “static” infrastructure to “dynamic” infrastructure. The challenges IT teams are addressing with the cloud operating model include:

1. Volume and distribution of services
2. Ephemerality and immutability
3. Deploying to multiple, heterogeneous target environments

These challenges force IT teams to adjust their approaches for each of the four layers of operation:

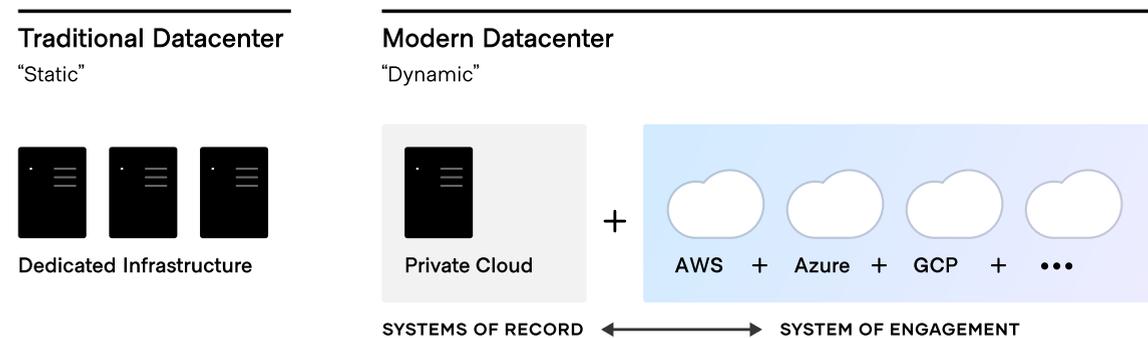
	Static		Dynamic
 Run	Dedicated Infrastructure	→	Scheduled across the fleet
 Connect	Host-based, Static IP	→	Service-based, Dynamic IP
 Secure	High trust, IP-based	→	Low trust, Identity-based
 Provision	Dedicated servers, Homogeneous	→	Capacity on-demand, Heterogeneous

Transitioning to a Multi-Cloud Datacenter

Organizations undergoing a digital transformation ultimately put pressure on teams delivering and supporting software applications to take advantage of the flexibility, speed, and agility of the cloud. Modern digital interactions are built cloud-first to provide fast, rich, personalized experiences informed by large-scale data processing and intelligence. Prioritizing digital interactions forces a change in the model for software delivery which is felt most by IT. Successful digital transformation depends on the IT team to build an organization-wide operating model for delivering cloud based applications. The core requirements of that model include infrastructure with:

- **Dynamic usage characteristics**, able to quickly scale loads up and down by orders of magnitude.
- **Support for fast development and iteration.** Many of these new systems may be ephemeral in nature, delivering a specific user experience around an event or campaign and then going away when no longer needed.

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



A common starting point for the cloud operating model is to enable the operations team to shift their focus away from provisioning dedicated servers based on homogenous sets of infrastructure to workflows that enable access to on-demand services from a variety of cloud and other service providers.

To address the core tenets affecting provisioning (volume and distribution of services, ephemerality and immutability, and deploying to multiple target environments), organizations are moving to an automation-based operating model for cloud infrastructure.

Cloud Operating Model: Minimizing Cloud Waste

The functionality of the cloud is simple enough: on-demand, pay only for what you use, and limitless capacity. But as cloud use grows, the cost of cloud can dig into the gross margins of the cloud providers' customers and thus requires more than refinement and optimization. [Public IaaS cloud spend was \\$64 billion in 2020](#), up 41% from the previous year, according to Gartner. With that much money on the line, organizations need to think about how to create and enforce policies to ensure operational consistency to manage costs in the cloud.

Research shows that an estimated [20% - 35%](#) of an organization's total cloud spending is wasted due to limited visibility, difficulty in enforcing policies, and tracking of cloud resources. The problem isn't new. In 2017, Gartner noted that "through 2020, 45% of organizations that perform lift-and-shift to cloud IaaS without optimization will be overprovisioned by as much as 55%, and will overspend by 70% during the first 18 months." In 2020, an estimated [\\$17.6 billion](#) was wasted on cloud and that number continues to increase as the cloud usage continues to grow.

Unnecessary cloud costs come in many forms:

Paying for unutilized cloud infrastructure

Many organizations moving to the cloud suffer from a lack of oversight, visibility, and tracking around what developers can provision, which can lead to idle resources, over provisioning, and orphaned resources. These unnecessary costs can quickly inflate your cloud bill.

Research from ParkMyCloud suggests that [nearly 45% of the enterprise cloud budget](#) is spent on non-production resources as companies continue to pay for resources that are sitting idle. Resources are typically paid for as if they were needed to run 24/7, but in reality they are often used mostly during a 40-hour workweek when for the other 128 hours of the week resources are sitting idle.

In addition, many organizations lack enforcement around provisioning practices, allowing operators to provision premium instances whether or not they are actually needed. For example, they may have no way to ensure that teams use a smaller instance during the development phase. Similarly, teams may be able to provision resources they end up never using. This often happens when teams still think in terms of buying physical gear and want to ensure they don't get caught short by unexpected demand.

Lastly, poorly monitored developers may contribute to the problem by failing to de-provision software or notify operations once they finish a project. This might be the result of developers simply forgetting

to de-provision, or the organization might have a long, complex process to get ops to de-provision unneeded resources.

One way to reduce these costs is to instill a culture of cloud-spending awareness within your organization. Unfortunately, this is often difficult to implement as organizations scale up and increase in complexity, and even when successful doesn't provide direct accountability. In a 2017 paper, [Ten Moves to Lower Your AWS IaaS Costs](#), Gartner noted that leveraging cloud elasticity by turning off resources when not in use via automation and by regularly rightsizing instances can save 74% or more.

Loss of developer productivity

Organizations have a strong need to standardize infrastructure deployment because cloud costs add up quickly when highly paid developers are unable to work efficiently. The goal is to unlock the full customer and business value offered by the cloud service providers, but that isn't always easy. In many cases, for example, each cloud service provides different tooling, forcing teams to build expertise in multiple systems.

There's also the struggle with navigating both on-premises and cloud infrastructure as organizations make the shift from static to dynamic infrastructure. 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 required to work with each technology and vendor. To create effective multi-cloud teams that maximize productivity and minimize waste, they need to apply their skills consistently regardless of the target environment.

Risks

It's no secret that the old-school, ticket-based gatekeeper IT I approach creates bottlenecks and limits developer productivity. In that kind of environment, policy isn't codified, but rather held tribally within the IT team. When policy is enforced, it happens manually by that gatekeeping organization, adding yet more work. But while this approach stymies the speed and developer self-service benefits that cloud infrastructure offers organizations, it can also help reduce risk.

As organizations move towards self-service infrastructure, many worry about incurring unforeseen costs due to possible security infractions. And limited visibility to infrastructure could promote cloud sprawl, shadow IT deployments, and new security challenges. In this situation, automating policy enforcement typically uses scans that occur after the infrastructure is provisioned. This opens up opportunities for risk while the out-of-compliance infrastructure is in place —and often in use — before it can be scanned.

Finally, some organizations take a least-common-denominator approach to policy enforcement, assuming that all infrastructure is susceptible to a limited set of known risks and checking only against those. This approach can leave open unique attack vectors that threaten bespoke or non-standard parts of the organization's infrastructure topology.

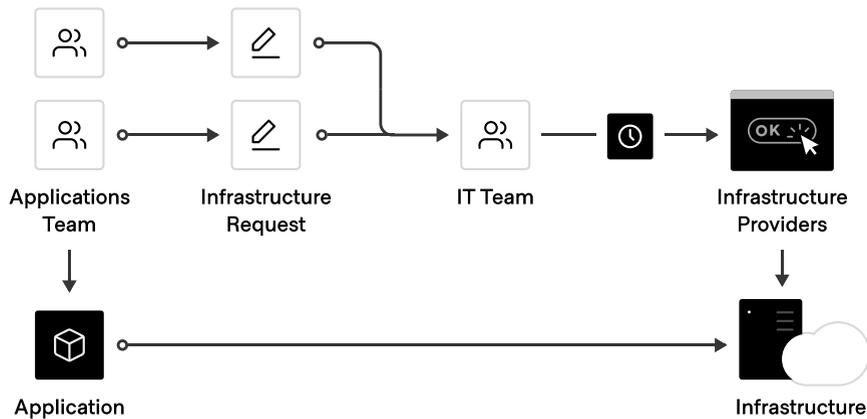
The challenge is even greater in multi-cloud environments, since many tools and workflows align to a single cloud vendor. A multi-cloud infrastructure can reduce productivity by requiring teams to master multiple workflows/APIs and extending lead times to support new lines of business. Having to secure, govern, and audit multiple workflows can also increase risk. But sticking to a single-vendor infrastructure topology can boost costs due to vendor lock-in and premium pricing stemming from a lack of competitive bidding.

Managing cloud costs is not easy, and cloud waste can seem inevitable. Fortunately, there are ways to minimize cloud waste.

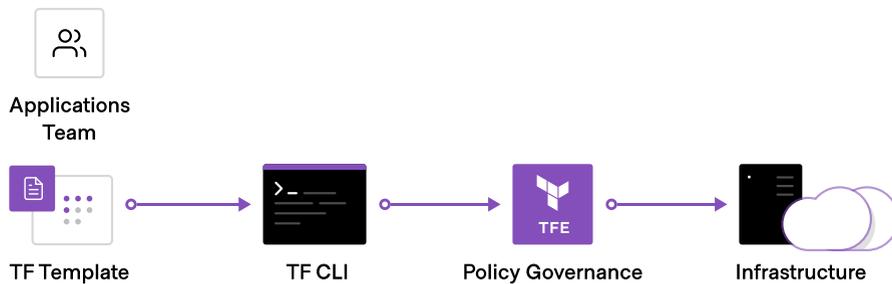
HashiCorp Terraform

HashiCorp Terraform, the world's most widely used cloud provisioning product, provides the foundation for cloud infrastructure automation using infrastructure as code for provisioning, compliance, and management in the cloud operating model. Terraform lets users leverage IaC to increase productivity, apply standardization and management policies, increase visibility into how infrastructure is set up and operating, and use self-service infrastructure to drive innovation.

Before Terraform



After Terraform



Using Terraform to codify infrastructure, increase infrastructure visibility and tracking can help organizations reduce their cloud spend by more than 20% by offering:

- Real-time control and proactive policy enforcement
- Reduction of unnecessary cloud resources such as idle resources, over provisioning, and orphaned resources
- Elimination of manual processes and bottlenecks
- Centralized management and control across technologies

Infrastructure as Code to Increase Productivity

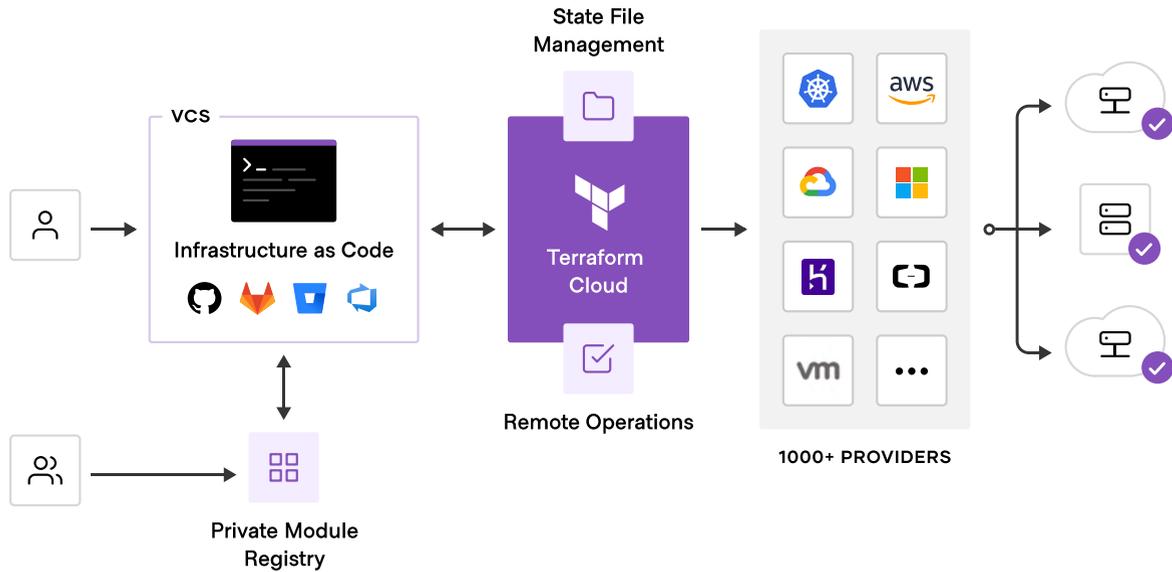
At the core of Terraform's solution to the above challenges resulting in cloud waste is [infrastructure as code](#). IaC allows you to provision and manage infrastructure with configuration files, rather than through a graphical user interface. That makes it easier to build, change, and delete your infrastructure in a safe, consistent, and repeatable way. Resources are codified in configurations that can then be integrated with version control systems (VCS) to make them easy to share, version, and reuse.

The IaC approach to cloud infrastructure automation enables organizations to use open source Terraform first for provisioning and then extend to Terraform Cloud and Terraform Enterprise for management in the cloud operating model. This can help reduce wasted spending as organizations grow their cloud footprint.

Let's take a closer look.

Version control systems. Many companies leverage the cloud management features of Terraform from within a pre-existing workflow or toolchain. Terraform enables this through integrations with major VCS, CI/CD, and service management tools as well as supporting a full REST API. These integrations help organizations drive operational consistency without impacting productivity.

Terraform users define infrastructure in a simple, human-readable configuration language called HCL (HashiCorp Configuration Language). Users can write their own unique HCL configuration files or borrow existing templates from the [public module registry](#). Most users store these configuration files in a VCS repository and connect that repository to a Terraform workspace. That connection lets users borrow best practices from software engineering to version and iterate on infrastructure as code, using VCS and Terraform Cloud as a delivery pipeline for infrastructure. [Terraform has integrations with Azure DevOps, BitBucket, GitHub, and GitLab.](#)



When you push changes to a connected VCS repository, Terraform will automatically trigger a plan in any workspace connected to that repository. This plan can be reviewed for safety and accuracy in the Terraform UI, and can then be applied to provision the specified infrastructure.

Standardization and Management to Minimize Risk

Organizations that use Terraform Cloud and Terraform Enterprise benefit from creating modules that serve as building blocks and enable them to manage 9,000 resources that amount to more than 60% of resources using Terraform. In addition, the creation of policies (as code) enables enforcement during the provisioning workflow to ensure modules are being used and policies are not being violated. Let's explore some of these Terraform functionalities in greater detail.

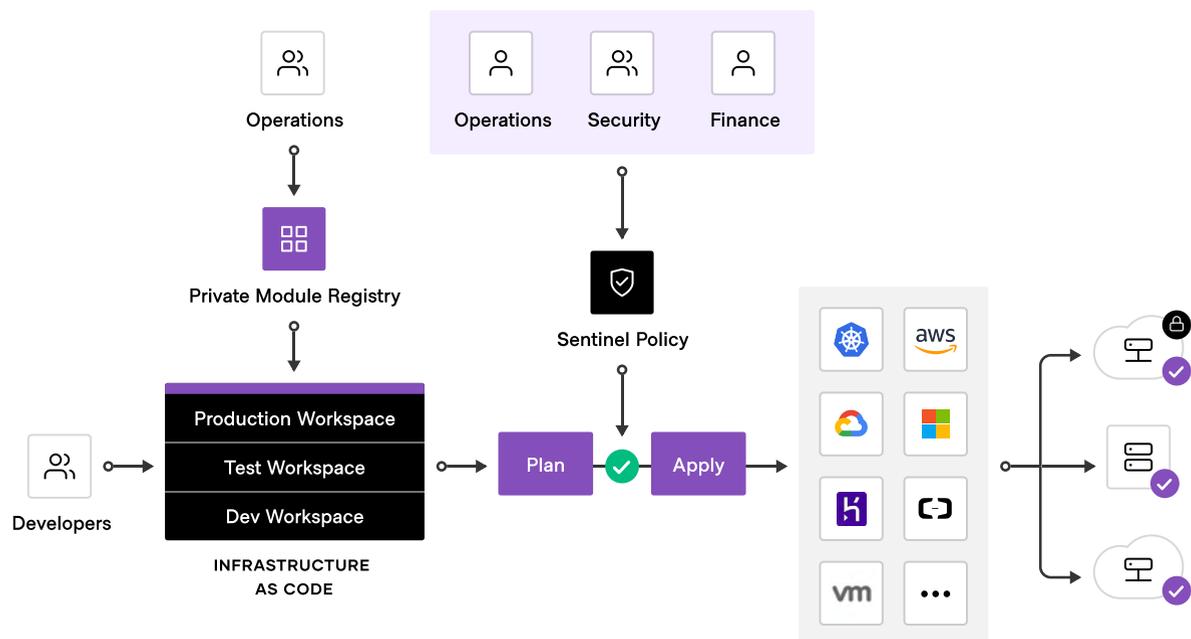
Modules. DevOps teams can create Terraform modules, which act as templates for deploying infrastructure within an organization. Modules can be designed according to organizational standards and industry best practices. Modules can include logic to consider instance types and sizes depending on whether a deployment is for production, development, or testing. Terraform Enterprise's private module registry then provides a centralized location for practitioners to filter and quickly deploy standardized, pre-approved modules across the entire organization.

Policy as Code. Terraform Cloud and Terraform Enterprise provide the foundation for multi-cloud infrastructure automation. They leverage infrastructure as code and policy as code for compliance and management to help control costs in the cloud operating model. With Terraform

Enterprise, organizations can leverage the cost benefits and flexibility of multi-cloud and teams can use a consistent workflow to provision, secure, govern, and audit any infrastructure to minimize unnecessary cloud resources, maximize productivity, and mitigate risk.

Cost management via policy enforcement. [Sentinel](#), HashiCorp’s policy as code framework, allows organizations to create cost-centric policies. Terraform then enforces Sentinel policies in the workflow on workspaces before it provisions them. Once an organization defines a guardrail in Sentinel, no infrastructure can be provisioned that breaks that guardrail — enforcement is automatic.

Administrators then have the ability to approve significant changes or to completely prevent specific workspaces from exceeding predetermined thresholds. For example, administrators could write policies that prevent large, expensive virtual machines from being provisioned unnecessarily, or enforce overall budget restrictions for a team’s deployments without prescribing what machines to use. Even further, policies can be dynamically written such that a certain threshold of change is allowed month over month, but nothing exceeding a certain dollar limit. Because Sentinel offers preventative, proactive policy organizations can confidently instill best practices for production workloads that help reduce cloud spend.



Visibility to Reduce Underutilized Resources

Not only does Terraform allow organizations to create and enforce operational best practices policies to manage costs and provide governance, Terraform provides a single source of truth by offering visibility into your infrastructure. Here's how Terraform provides infrastructure visibility through estimations and audit logs:

Cost estimation. One of the biggest benefits of cloud infrastructure is pay-as-you go pricing. But it can be challenging to understand the cost implications of new or changed infrastructure before it is provisioned and applied. Most organizations rely on after-the-fact alerts from their cloud provider, using dedicated third-party services that continually monitor changes in cloud cost, or simply wait until they receive their end-of-the-month bill to see how their changes affect their costs. Terraform's cost-estimation capability programmatically estimates the cost of new cloud deployments or changes to existing ones, before applying those changes or actually incurring the costs.

Audit logging. Terraform Enterprise offers rich [audit logging](#) capabilities for organizations that need insight into the resources managed by Terraform. Audit logs capture information whenever any resource managed by Terraform Enterprise is changed, so teams can understand what changes were made and who made them. Many organizations leverage audit logging to achieve and ensure regulatory compliance.

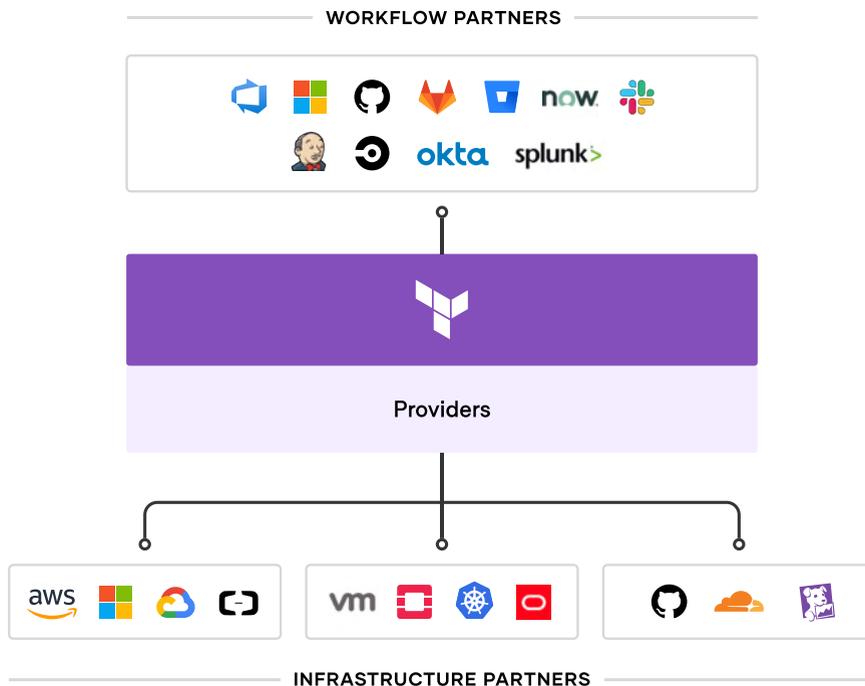
Self-Service Infrastructure to Drive Innovation

Through integrations with top CI/CD pipelines and ITSM systems, Terraform provides a method for developers to reuse modules, collaborate to increase efficiency, and provision their own self-service infrastructure without an operator bottleneck, increasing operator productivity and developer agility. With Terraform, organizations can move beyond ticket-based gatekeeper systems to position central IT services as a self-service enabler of speed, without letting costs get out of hand.

Continuous integration/Continuous delivery (CI/CD) pipeline. Terraform can be called from within most popular CI/CD pipelines, including Jenkins, Circle, and Travis. Many users leverage Terraform's programmability to automate as much of their provisioning workflow as possible, while enforcing guardrails through policy as code. Terraform's API-driven run provides flexible provisioning workflows using an infrastructure as code approach that any organization can manage.

A continuous integration (CI) system monitors changes in Terraform code and drives provisioning using Terraform Cloud's REST API. This approach lets organizations implement a range of actions in their CI pipeline as part of an infrastructure provisioning workflow and still benefit from Terraform Cloud capabilities such as private modules, state management, [policy as code](#) (Sentinel) and more.

IT Service Management (ITSM). Our newest integration connects the human workflow power of ServiceNow with the infrastructure workflow capabilities of Terraform Enterprise. Terraform Enterprise also includes a first-class ServiceNow integration. [ServiceNow](#) provides digital workflow management, helping teams work quickly and efficiently with one another by offering a straightforward workflow for their interactions. The ServiceNow Service Catalog offers a storefront of services that can be ordered by different people in the organization. One common request between teams is for cloud resources: perhaps a developer needs a fleet of machines to test out a codebase or the IT team in finance has a request for infrastructure to run their new accounting software. For organizations that use the ServiceNow Service Catalog, the requests can be submitted through ServiceNow and routed to the right team for cloud infrastructure. This enables teams that are not code-centric to safely adopt best-in-class provisioning workflows and tooling, while still developing competency with infrastructure as code.



Conclusion

As organizations begin to adopt the cloud operating model, they face the challenge of provisioning cloud infrastructure as well as high cloud costs. Many organizations worry that even if the cloud unlocks new speed and agility, it also creates the possibility of out of control spending.

HashiCorp Terraform offers a powerful solution by combining an infrastructure as code approach to provisioning with a policy as code approach to cost control. This enables organizations to have both high agility and high control as they develop competency in infrastructure provisioning and management of their cloud costs. And as your organization's cloud adoption grows, so does Terraform's ROI.

About HashiCorp

HashiCorp is the leader in multi-cloud infrastructure automation software. The HashiCorp software suite enables organizations to adopt consistent workflows to provision, secure, connect, and run any infrastructure for any application. HashiCorp open source tools Vagrant, Packer, Terraform, Vault, Consul, and Nomad are downloaded tens of millions of times each year and are broadly adopted by the Global 2000. Enterprise versions of these products enhance the open source tools with features that promote collaboration, operations, governance, and multi-data center functionality. The company is headquartered in San Francisco and backed by Mayfield, GGV Capital, Redpoint Ventures, True Ventures, IVP, and Bessemer Venture Partners. For more information, visit www.hashicorp.com or follow HashiCorp on Twitter [@HashiCorp](https://twitter.com/HashiCorp).

