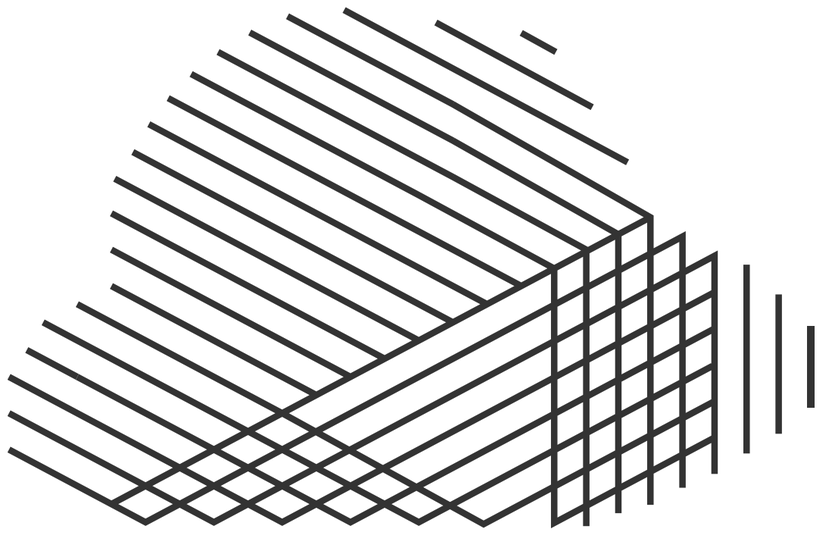
HashiCorp



Cloud Adoption

DevOps and the realities of multi-cloud

Introduction

Organizations of all sizes are adopting the cloud operating model for their application workloads. Whether to optimize the costs of running and managing their data centers or to enable development teams to efficiently build the applications of tomorrow, the majority of organizations are interested in the benefits of the cloud model.

While early cloud adoption was largely about building new applications on Amazon Web Services (AWS), today it is clear that the cloud model is not about just one cloud but the ability to embrace multiple clouds. The continuing investments made by Microsoft Azure, Google Cloud Platform, Oracle Cloud, IBM Cloud, Alibaba Cloud, and VMware provide compelling infrastructure alternatives, each with their own unique value propositions. These cloud providers will continue to build their portfolio of application services over the next decade.

The cloud model—characterized by infrastructure heterogeneity, on-demand self-service infrastructure, and the lack of a clear network perimeter where physical server hosts provide the traditional basis for networking—is likely to represent the way we think of infrastructure for the next several decades.

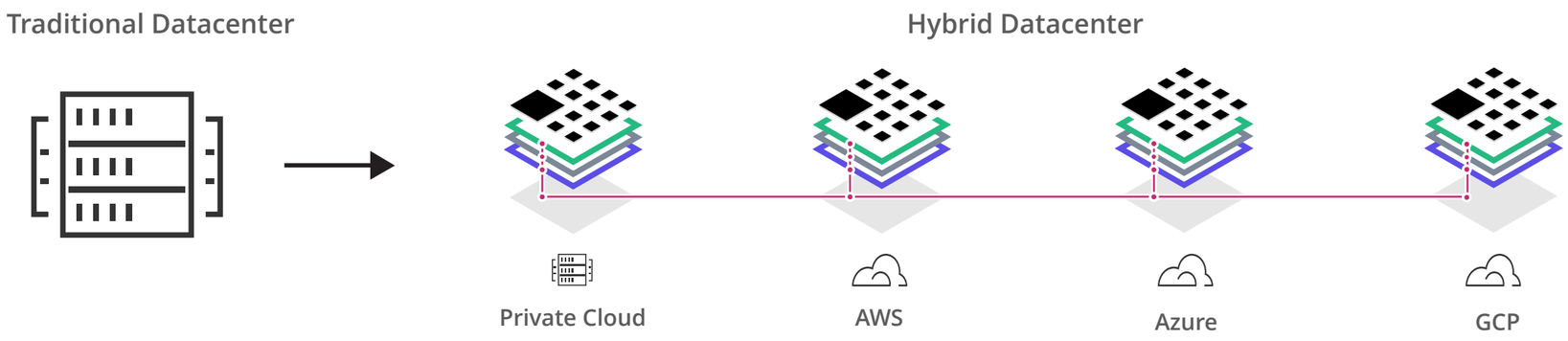
Similar to infrastructure transitions of the past—Mainframe to Client-Server in the ‘80s or Client-Server to Web in the ‘90s—the transition to cloud introduces a very different operating model for IT as well. The cloud operating model requires navigating the transition from a relatively static pool of homogeneous infrastructure in dedicated data centers with physical hosts and protected by a strong network perimeter, to a distributed “fleet” of dynamic infrastructure that is provisioned as-needed and with no notion of a physical ‘host’.

Below are the major changes in the operating model as infrastructure shifts from static to dynamic. Each one of these areas will be explained in more detail later in this paper.

	Static Data Center	Dynamic Cloud
Run	- Dedicated instances per application - Under utilized machines	- Decouple instances from applications - Bin packed machines
Connect	- Load balancers - Firewalls - Configuration Management	- Service discovery - Service segmentation - Immutable infrastructure & runtime configuration
Secure	- Hardcoded secrets - IP as the source of identity - HSMs for encryption	- Dynamic secrets - Multiple user & application identity sources - Software encryption
Provision	- Pre-provisioned resources - Hardware and CapEx oriented - Single provider	- Resources provisioned on demand - Software and OpEx oriented - Multiple providers

The Challenge

The primary challenge of cloud adoption is that all four core constituents in IT—operations, security, networking, and development teams—must each internalize the implications of the cloud model.



Core Constituents in IT

We believe a practical place to begin is by considering the core roles in IT and then deconstructing the core challenges that they will need to address as they adopt the cloud model.

This separation of concerns allows us to decompose the challenges faced by most organizations into smaller well-defined problem sets that provide a practical path.

Unique Challenges for Operations Teams

In the traditional data center, a limited number of virtualized, homogenous servers are available to the operations team. Operators then provision compute capacity across this pool of physical servers and make it available to individual teams to deploy their respective applications.

By contrast, the scale of cloud infrastructure is essentially infinite: the server fleet really has no practical limit since the cloud providers operate at enormous scale. Each cloud provider offers a unique inventory of available services: a compute node on AWS, for example, is subtly different from a compute node on Azure or Google Cloud Platform. While this heterogeneity is what gives each cloud its richness, it also introduces tremendous complexity for operations professionals who will need to learn the idiosyncrasies of provisioning each cloud, infrastructure, and external service to leverage its unique value. The core challenges then for operations teams are:

Managing Scale and Elasticity

The traditional operating model has been to provision virtual machines upon which applications will subsequently be deployed. The cloud model assumes that infinite quantities of infrastructure will be provisioned on-demand by many teams and then destroyed when no longer required.

Since cloud providers expose their services using APIs, operations teams are adopting the notion of infrastructure as code. In this manner, an infrastructure topology can be codified as a simple script and then executed each time that infrastructure is required. By changing simple elements of the code, for example ‘n = 1’ to ‘n = 1,000’, practically infinite amounts of identical infrastructure can be provisioned instantaneously.

Heterogeneity

An infrastructure as code approach enables reusability, but the reality is that large organizations will ultimately utilize more than one cloud provider in addition to their own private on-premises data center. This heterogeneity is inevitable as organizations address business drivers including data sovereignty, regional redundancy, application-specific runtime environments, or on-boarding new branches of the business in new countries.

Because each infrastructure provider has a unique provisioning model, a challenge for operations teams is to determine a strategy that gives them a consistent provisioning workflow regardless of infrastructure type while leveraging the unique capabilities of each cloud provider.

In addition to core compute capacity, most infrastructure teams will need to automate the provisioning of non-cloud components in their blueprints. This might take the form of monitoring agents, Application Performance Management (APM) products, or other software components required by the application teams.

Managing this heterogeneity while also provisioning infrastructure on demand for the application teams is exponentially difficult as both the number of teams and the infrastructure providers they use grows.

Infrastructure delegation

Private infrastructure consisted of homogeneous, static infrastructure that was provisioned for long periods of time and dedicated to specific users. Cloud now consists of heterogeneous, dynamic infrastructure that is provisioned on a larger scale more frequently, and for many users to consume on-demand. The challenge is for operators to decompose and assign ownership of work for these large scale cloud infrastructures and then to provide well-defined templates of infrastructure configurations that can be consumed on-demand by other operators and developers.

Unique Challenges for Security Teams

The traditional data center had a clear, ‘four walls and a pipe’ network perimeter. Firewalls serve as bulkheads between frontend, user-facing applications and backend databases. Anyone inside the network is assumed to be authorized to access the infrastructure. IP addresses are generally static, which allows security professionals to provide additional constraints on application interactions based on IP addresses.

For organizations embracing cloud, the scenario for security has changed quite a bit. Cloud doesn’t have a distinct perimeter. With multi-cloud, the surface area security teams are concerned with protecting expands exponentially. A lack of control over network topologies makes it hard to force all traffic through security or compliance tools. Network topology is software-defined, and any server can become Internet-facing with a few API calls. Infrastructure may also span multiple sites, meaning there isn’t a single ingress point to allow secured traffic to flow into a network.

The decomposition of monolithic applications into highly ephemeral microservices means that IP addresses are highly dynamic, rendering IP-based security inappropriate for many scenarios.

The core challenges then for security teams are to rethink the core assumptions around the network perimeter while enabling development and infrastructure teams to adopt cloud for their new application workloads. We must consider the reality of low-trust networks and focus on protecting the content of the applications themselves.

Identity as the basis for access management

The dynamic nature of cloud infrastructure implies that the traditional model of access management between applications and application components based on the IP address of the requestor can no longer apply.

Instead, security teams will need to utilize a different trust model to enforce systems and application access, and identity is the logical choice. Identity models are well established in every organization and therefore available for use in systems and applications as well. The ability to leverage a trusted source of identity to enforce system or application access serves as a logical replacement for IP address in the cloud model.

Managing secrets

Secret sprawl—development and operations teams leaving credentials easily accessible inside the network—is endemic in most organizations since teams have safely assumed that the network perimeter is secure. As we transition to cloud, the lack of a clear network perimeter implies that the network is inherently low-trust and therefore there is considerable risk if application secrets are treated in this way.

A centralized approach to secrets management, therefore, is an absolute prerequisite to the application delivery process in the cloud model. In this manner, security teams must determine a way to allow operations and development teams to both provision and deploy onto cloud infrastructure while providing a safe mechanism for secrets management.

Encryption

If we make the assumption that networks can no longer be secure, then encryption of data inside the network is a logical step. Security teams should encrypt application data both at-rest and in-flight but do so in a way that does not compromise the application delivery velocity.

Unique Challenges for Networking Teams

The networking challenge is perhaps the most difficult in the cloud model. In the traditional static data center, all networking is grounded in the notion of physical servers (“hosts”): the basis of networking is connecting those hosts and everything that resides on them together.

However, in the cloud model there is no concept of permanent hosts. Instead we have dynamic pools of infrastructure (often with transient containers atop them) that are used for a period of time and then destroyed when no longer needed.

The cloud approach then is to think about infrastructure and applications not in terms of hosts but as services instead. There are simply infrastructure services (a server node or VM) and application services (a microservice, for example, containing application logic). This requires networking teams to reconsider their challenges across the three essential elements of networking:

Connectivity

Rather than discovering and connecting hosts, we need to think about how to discover and connect services—a particularly acute challenge given the ephemeral nature of infrastructure in the cloud model.

For development teams this is a familiar concept: they are accustomed to using a service registry to store and discover the latest version of application services required by their application (such as reusable application elements). The cloud model is no different: a new kind of service registry is required that is appropriate for the new dynamism and scale.

Operations teams in the cloud model then must adopt a dynamic registry that provides a shared and real-time understanding of services in use at any time. This provides the core inventory of components that can then be connected.

Configuration

Once a clear view of the services is available, networking teams will need to use that registry to configure those services. While in the traditional model, an administrator can SSH into a machine or device to update configuration, but the lack of static IP addresses in the cloud model confounds this tactic. Instead, networking teams must find a way to dynamically update the distributed fleet of services in a different way.

Security

Services move across hosts, hosts are frequently created and destroyed, and services no longer have long-lived IPs. While in the traditional model, networking teams define a clear network perimeter and then protect that with a firewall. The dynamic nature of the new cloud model makes this perimeter-based approach difficult to scale as it results in complex network topologies and a sprawl of short-lived firewall rules.

In the low-trust cloud network environment, networking teams will need to utilize service segmentation to enforce security. In this way they can configure rules—for example, “Application_Server_A” can connect to “Database_B”—that are specific to a service rather than to its IP.

Unique Challenges for Development Teams

The adoption of cloud is often driven by new application requirements and by developers pushing a shift to the ‘DevOps’ model. The runtime layer for their applications, however, is inevitably different from previous models: rather than deploying an artifact to a static Application Server, developers are now deploying their application artifacts (often packaged into a container) to an application scheduler atop a pool of infrastructure provisioned on demand.

The challenges then for teams running this layer of infrastructure in the cloud model are:

Separation of concerns

Developers are now able to better focus on the application artifact itself, rather than a specific server, without regard for the infrastructure specifics. Instead, they deploy their application to a pool of compute that is configured in a predictable way (either one node or one million) to run that application.

The challenge then is a model that allows for this clear separation of concerns and abstracts them from a detailed knowledge of the underlying infrastructure: simply provide the application artifact together with instructions for how much infrastructure it needs, and delegate the rest.

Bin packing

Because the application artifact will be deployed to a pool of compute that charges by the hour, it is critical that the core scheduling of compute, I/O, and storage consumption is as efficient as possible. Therefore, the cloud model introduces the need for an application scheduler—and in particular, one that can schedule all aspects of the application, regardless of type. It is this heterogeneity of application type that is critical so that all aspects of the application, including non-container formats, can be scheduled efficiently.

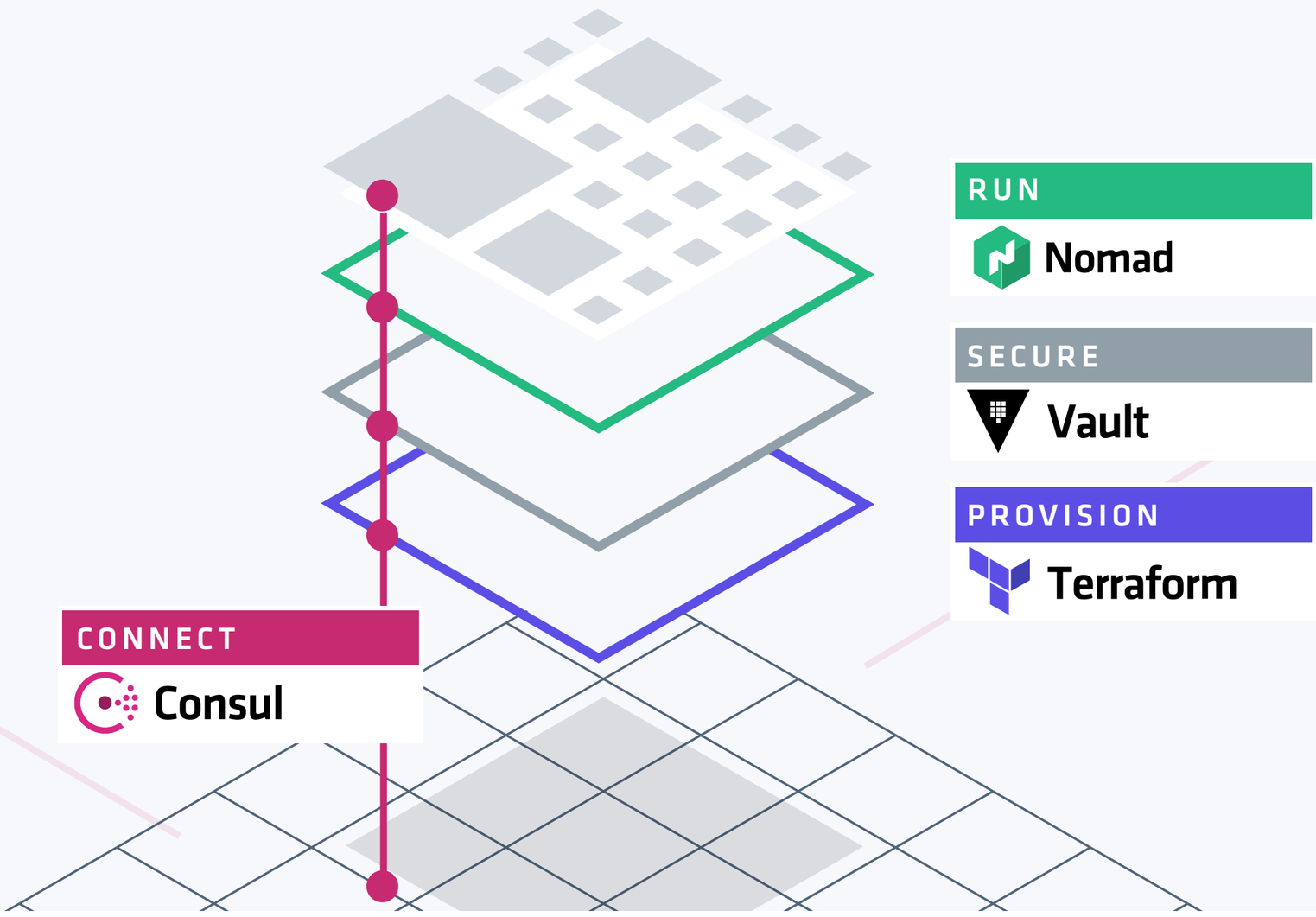
The shift to the cloud model has significant implications for all aspects of IT: operations, security, networking, and development teams. As was the case with prior infrastructure transitions, these teams must internalize the changes of this model.

Cloud Infrastructure Defined

A consistent consistent approach to provision, secure, connect, and run any infrastructure for any application

Organizations can address these challenges of cloud adoption with tools that provide a consistent workflow to a single, well-scoped concern at each layer of the infrastructure stack. This focus on workflows over technologies allows underlying technologies to change while the workflow for each part of the organization does not. As a result, organizations simplify challenges related to their diversity of technology.

HashiCorp provides a suite of products that address the challenges of each constituent as they adopt cloud. Each tool addresses a focused concern for the technical and organizational challenges of infrastructure automation, so tools can be adopted one at a time or all together.



HashiCorp

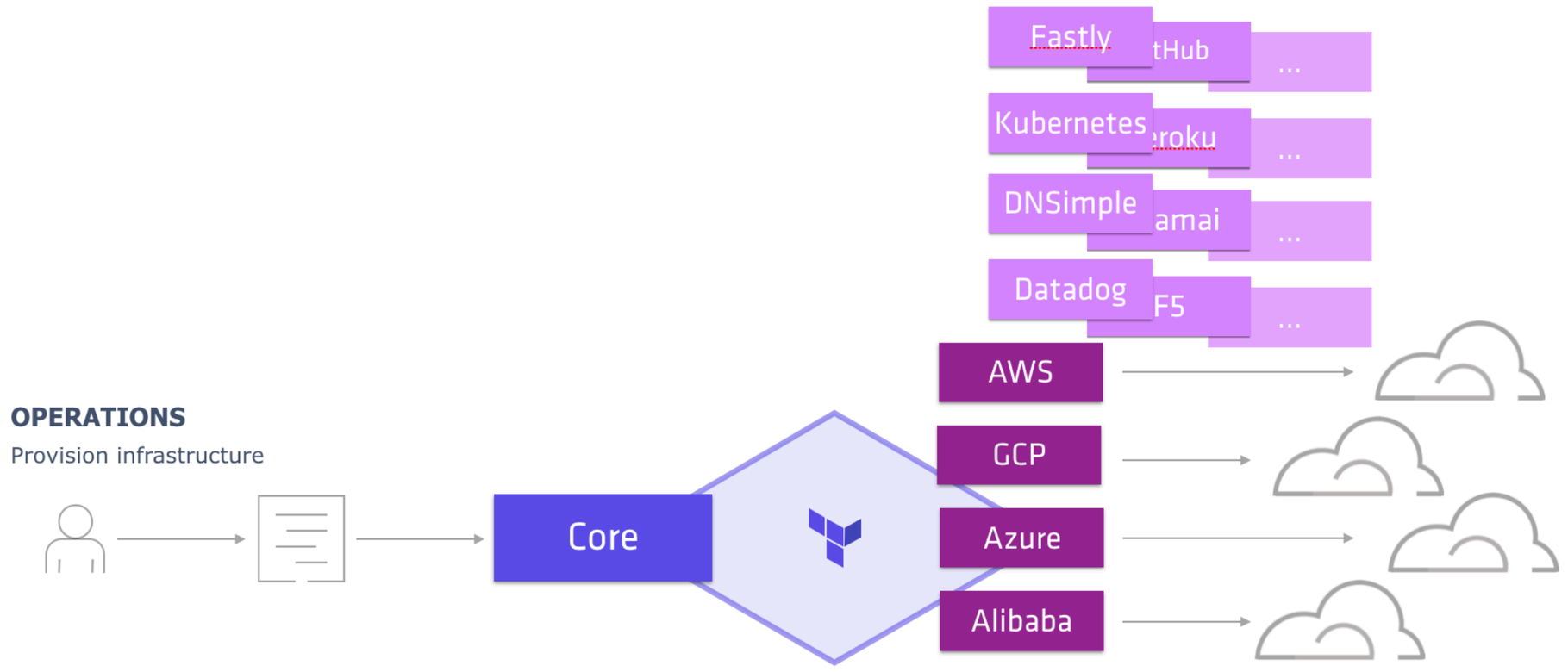
Terraform

LEARN MORE ABOUT TERRAFORM

There are two parts to Terraform's extensible architecture: Terraform Core, which is the core of the product, and then a series of Providers—plugins to support cloud types such as AWS, GCP, Azure, and vSphere. In this way, a user can adopt a common provisioning workflow and then apply that to any infrastructure type. These Providers are endorsed and in many instances in fact developed by the cloud providers themselves who view Terraform as the lingua franca for cloud provisioning and so reducing a barrier to adoption of their platforms.

In addition to support for provisioning on the major cloud providers, Terraform also supports more than 100 infrastructure types (each with their own Terraform Provider), and 1,000 unique resource types. The open source nature of the Providers makes it easy for anyone to contribute and improve providers as the infrastructure vendors add new capabilities.

Operators codify infrastructure in the form of Terraform templates, which typically combine infrastructure types (for example, Datadog agents, or the Kubernetes cluster in addition to AWS services). By applying the infrastructure as code concept, operators can collaborate and share these templates in GitHub (or other version control systems) and follow the same principles that software developers use to collaborate on code.



Terraform can be used by a small number of operators to produce Terraform templates that can be consumed by a large team of developers. This producer/consumer relationship is a key ingredient to unlocking the organizational challenge of cloud adoption because it reduces the friction and bottlenecks of infrastructure provisioning while giving control to the operations team to define, govern, and standardize the templates.

Provisioning requires infrastructure-specific images because there is no common packaging format for virtual machines across providers. HashiCorp Packer enables operators to build many machine image types from a single source. A Terraform configuration can reference these images to provision infrastructure using the cloud-specific images created by Packer.

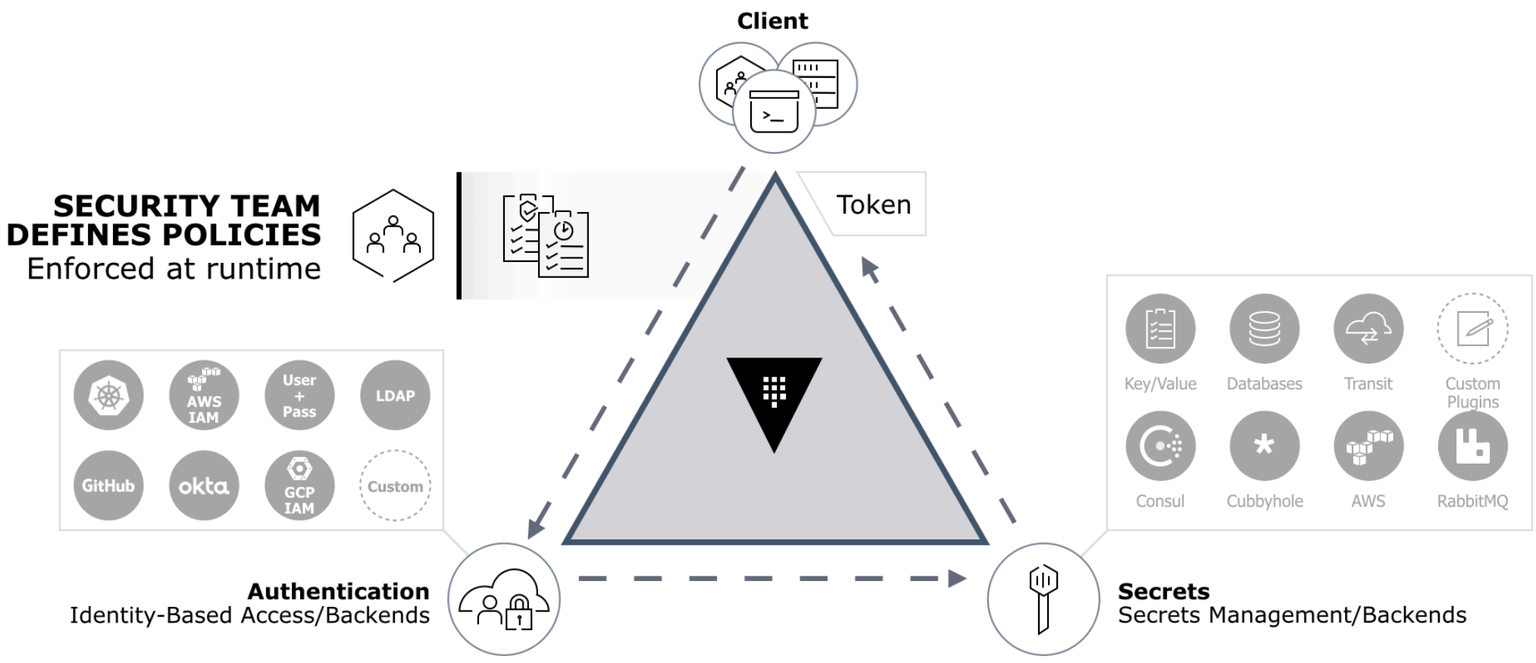
HashiCorp

Vault

LEARN MORE ABOUT VAULT

Vault addresses the challenge of security for distributed application infrastructure in the low-trust network model of cloud. At its core, Vault allows users to *leverage any trusted source of identity to enforce systems and application access*.

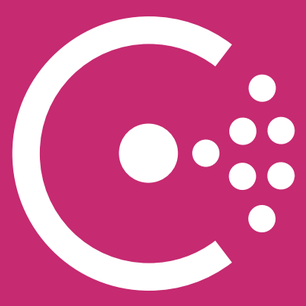
The core model is described below:



With support for all of the primary identity models (Active Directory or LDAP on-premises and the cloud native IAM models for each infrastructure type), Vault provides a single mechanism to bridge identity across these different platforms. Users can adopt a common workflow regardless of identity model.

By providing a centralized approach to secrets management and rotation (e.g., private encryption keys, API tokens, and database credentials), Vault also helps to address the organizational reality that security teams must control and define policy related to secrets management, without becoming a bottleneck to the operations and development teams. Its extensible architecture provides support for many types of storage and authentication systems, and its policy support provides granular access control between human and server or between server and server.

Security teams can optionally use Vault to encrypt data at rest and in transit without any modifications to the application itself.



HashiCorp

Consul

LEARN MORE ABOUT CONSUL

Consul delivers a distributed service networking layer to connect, secure, and configure services in a dynamic infrastructure. It addresses the three key challenges of networking in the new cloud model: connectivity, security, and run-time configuration.

Consul provides a common registry of all infrastructure and application services in the environment in real-time. This registry can then be used as the shared system of record in the distributed fleet and allows users to then connect, configure and secure those services.

For example, a web server (an example of an “infrastructure service”) can use Consul to discover and connect to its upstream database or API services. Or rather than hard-coding a network address, a developer can push the discovery of dependent services into the application runtime. A running service broadcasts its availability and can then be easily reached by other applications.

Consul then enables security and networking teams to allow or deny service communications with a set of simple intentions. For example, a web server can be allowed to connect to a database. TLS certificates are used to identify and secure communications. Developers can use Consul’s built-in proxy to automatically establish inbound and outbound TLS connections between services without rewriting applications.

And in addition to providing connectivity and security for all services, Consul enables operators to dynamically configure those services. For example, while an application is running, Consul can monitor and flag degraded instances, while directing traffic to healthy instances and notifying developers or operators for any issues.

This distributed networking layer that supports infrastructure and developer services is a foundational element for enabling the cloud model.

HashiCorp

Nomad

LEARN MORE ABOUT NOMAD

Nomad is a multi-datacenter-aware cluster manager and scheduler. It provides a consistent approach for deploying any application. This includes batch, dispatch, and long-running services:

- Batch workloads include big data applications that need jobs to complete quickly.
- Dispatch workloads include short-lived, elastic applications.
- Long-running services need secure and highly available data centers.

Developers codify the runtime requirements for applications in a declarative configuration file. Nomad uses this file to place the application across a fleet of machines. Machines may be in a single cloud, span many geographic cloud regions, or be in many clouds.

Infrastructure operators provision the fleet of machines, whereas developers use Nomad to handle the deployment across that fleet of machines. In this way, we decouple infrastructure provisioning from application deployment.

Accelerating Cloud Adoption

We've described a consistent toolset to empower operators and developers to provision, secure, connect, and run any infrastructure for any application.

It's important for organizations to be able to quickly and efficiently run applications and infrastructure on the cloud best suited for their needs, while still retaining flexibility in their choice as applications and cloud offerings evolve.

This is the fundamental purpose of the HashiCorp suite—to provide customers with the infrastructure automation capabilities they need as they move to cloud. The "lego piece" approach of HashiCorp software allows organizations to incrementally adopt the tooling they need and integrate with their existing systems.

Read about the latest feature updates for Terraform, Vault, Nomad, and Consul on the HashiCorp Blog.

Updated: 07/31/18