HashiCorp | Microsoft Azure

Unlocking the Cloud Operating Model

Achieving the fastest path to value
in a modern, hybrid-cloud datacenter

WHITEPAPER

Unlocking the Cloud Operating Model on Microsoft Azure

HashiCorp and Microsoft talk to organizations of all sizes on how their infrastructure needs to evolve and how they're adopting the cloud operating model in order to accelerate automation. For most enterprises, digital transformation means delivering new business and customer value at scale, efficiently. The implication for Enterprise IT is navigating the shift from cost optimization models to speed-optimization models. The cloud is an inevitable part of this shift as it presents the opportunity to rapidly deploy on-demand services with limitless scale. To unlock the fastest path to value of the cloud, enterprises must consider how to industrialize the application delivery process across each layer of the cloud embracing the cloud operating model, and tuning people, process, and tools to it.

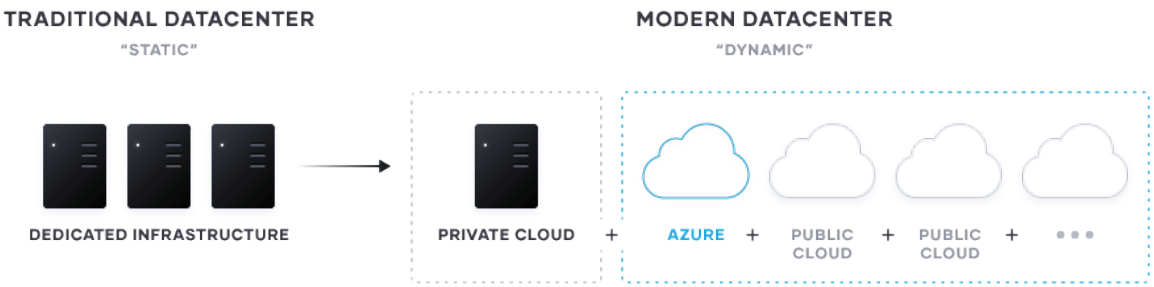
This whitepaper looks at four specific areas across infrastructure, security, networking, and application delivery on Microsoft Azure.

Setting the stage: The shift from a static to dynamic environment

The transition to cloud, and hybrid-cloud, environments is a generational transition for IT. This transition means shifting from largely dedicated servers in a private datacenter to a pool of compute capacity available on demand. The cloud presents an opportunity for speed and scale optimization for new "systems of engagement" - the applications built to engage customers and users. These new apps are the primary interface for the customer to engage with a business and are ideally suited for delivery in the cloud as they tend to:

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

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



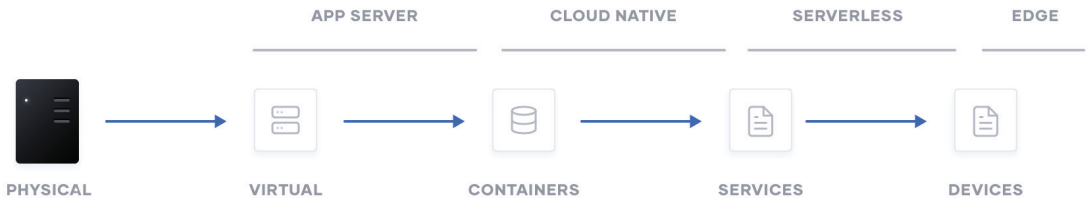
Unlocking the Cloud Operating Model on Azure

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

When working with customers on Microsoft Azure, cloud implementation is an iterative process of migrating and modernizing the digital estate, coordinated with targeted business outcomes and change management controls. During each iteration, workloads are migrated or modernized in alignment with the strategy and plan. Decisions regarding IaaS, PaaS, or hybrid are made during the assess phase of the Migrate methodology to optimize control and execution. Those decisions will drive the tools used during each iteration of the migration phase within the same methodology.

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

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



For cloud computing to work, there needs to be consistent workflows that can be reused at scale across multiple cloud providers. This requires:

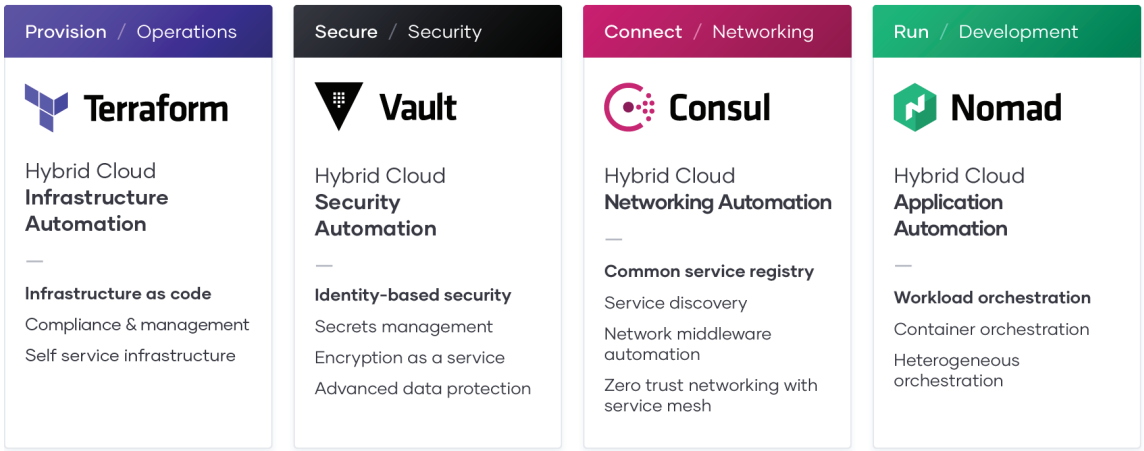
- Consistent instruction sets for provisioning
- Service-based networking for applications
- Identity-based access management

Implications of the Cloud Operating Model

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

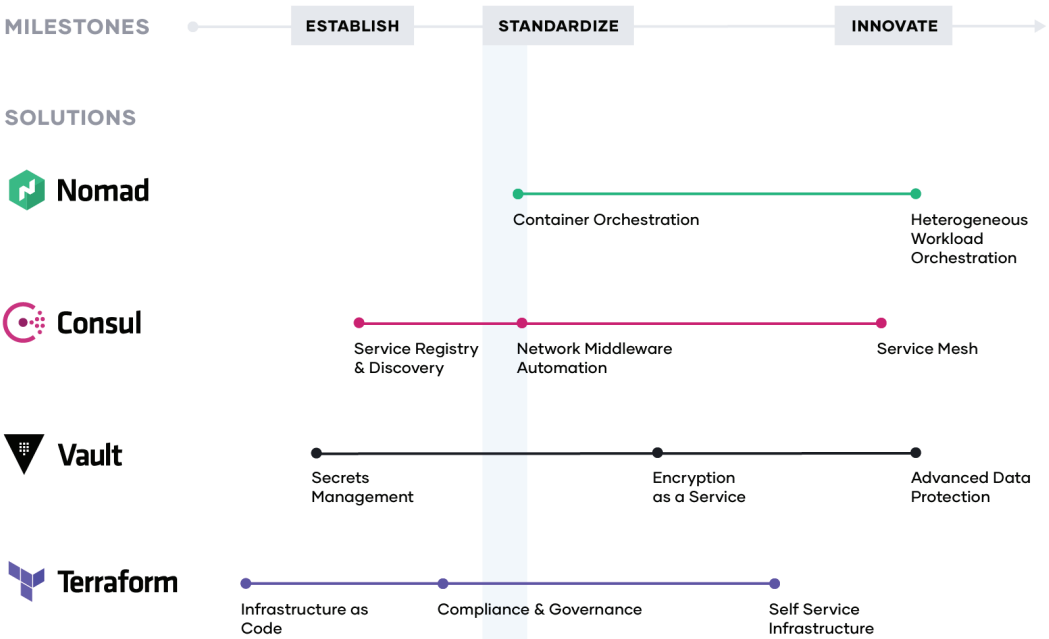
	STATIC		DYNAMIC
Run	Dedicated Infrastructure	→	Scheduled across the fleet
Connect	Host-based Static IP	→	Service-based Dynamic IP
Secure	High trust IP-based	→	Low trust Identity-based
Provision	Dedicated servers Homogeneous	→	Capacity on-demand Heterogeneous

EXPANDING USE OF THE HASHICORP STACK INCREASES MATURITY AND VELOCITY FOR OUR CUSTOMERS



Together, Microsoft and HashiCorp tools come together to help IT and the business units align on a clear strategy and plan to guide implementation activities. As teams deliver on each shared service for the cloud operating model, IT velocity increases. The greater cloud maturity an organization has, the faster its velocity.

EXAMPLE ENTERPRISE JOURNEY TO UNLOCK A CLOUD OPERATING MODEL



As customers unlock the cloud operating model, the journey typically includes three major milestones:

Establish the cloud essentials - As you begin your journey to cloud, the immediate requirements are provisioning and securing your infrastructure. This is done through adopting infrastructure as code and implementing a secrets management solution, respectively. These are the bare necessities that will allow you to build a scalable and truly dynamic cloud architecture that is futureproof.

Standardize on a set of shared services - As cloud consumption starts to pick up, you will need to implement and standardize on a set of shared services so as to take full advantage of what the cloud has to offer. This also introduces challenges around governance and compliance as the need for setting access control rules and tracking requirements become increasingly important.

Innovate using a common logical architecture - As you fully embrace the cloud and depend on cloud services and applications as the primary systems of engagement, there will be a need to create a common logical architecture. This requires a control plane that connects with the extended ecosystem of cloud solutions and inherently provides advanced security and orchestration across services and multiple clouds.

Infrastructure automation with Terraform

Terraform and Azure: Infrastructure as code

Over the last five years, HashiCorp and Microsoft have partnered closely in the development of Terraform. Together, we are focused on empowering organizations with the scalability and flexibility of infrastructure as code (IaC). HashiCorp Terraform codifies infrastructure in configuration files that describe the topology of cloud resources. These resources include virtual machines, storage accounts, and networking interfaces. The Terraform CLI provides a simple mechanism to deploy and version the configuration files to Microsoft Azure. This is truly a hybrid and multi-cloud implementation that connects a customer's private cloud infrastructure using Hyper V, as we all as their public-cloud deployment with Azure.

Organizations need the right tools in order to empower engineers to focus on problem-solving and building, rather than submitting and waiting on provisioning requests. Further, teams need to deploy the same code to multiple regions and environments in a consistent, safe manner.

With Terraform, you can provision environments in under an hour. Applications move from [dev-test] to production. Environments are easily reproducible and there is no risk from patching because you can test exact infrastructure templates.

Additional benefits of Terraform on Azure

Better Together: HashiCorp and the community
Integrating with GitHub helps realize the value of version-controlled infrastructure with Terraform. In addition to providing a single, familiar view where Terraform users can see the status and impact of their changes, the integration also brings about continuous integration and testing for infrastructure changes. The consistent GitHub workflow pairs well with HashiCorp's goals of providing a technology-agnostic workflow for provisioning, securing, and running any infrastructure for any application. For more, explore the GitHub repository (github.com/terraform-providers/terraform-provider-azurestack).

Enterprise-ready
Streamline operations and provision any infrastructure more securely and efficiently with Terraform Enterprise. Centralize infrastructure deployment within one workflow and provision, govern, and audit any environment.

Leveraging Terraform on Azure

Leveraging Terraform on Azure empowers you to gain flexibility, security, and collaboration across your organization.

Automate infrastructure management

Terraform's template-based configuration files enable you to define, provision, and configure Azure resources in a repeatable and predictable manner. Automating infrastructure has several benefits:

- Lowers the potential for human error while deploying and managing infrastructure.
- Deploys the same template multiple times to create identical development, test, and production environments.
- Reduces the cost of development and test environments by creating them on-demand.

Understand infrastructure changes before being applied

As a resource topology becomes complex, understanding the meaning and impact of infrastructure changes can be difficult. The Terraform CLI enables users to validate and preview infrastructure changes before application. Previewing infrastructure changes in a safe manner has important benefits:

- Team members can collaborate more effectively by quickly understanding proposed changes and their impact.
- Unintended changes can be caught early in the development process.

Deploy infrastructure to multiple environments on-premises or in the cloud

Terraform is adept at deploying an infrastructure across multiple cloud and on-premises providers. It enables developers to use consistent tooling to manage each infrastructure definition.

Learn how to use Terraform to reliably provision virtual machines and other infrastructure on Azure with the Terraform on Azure documentation found at: docs.microsoft.com/en-us/azure/terraform.

Building a zero-trust security model with Vault and Azure

Secrets, encryption, protection

Most organizations today have the issue of secrets sprawl. These secrets include database passwords, certificates, and private keys that should be constantly protected. However, this should not impact the speed and reliability with which code is shipped. Working with Microsoft, HashiCorp launched Vault with a number of features to make secrets management easier to automate in Azure cloud.

Vault offers a wide array of Secrets Engines that go far beyond just basic K/V management. Vault Secrets Engines can manage dynamic secrets on certain technologies like Azure Service Principles, Databases, and Datastores. These secrets are both time and access bound, which often eliminates the need to rotate secrets. Dynamic secrets help reduce the blast damage of any leaked secrets or compromised systems because every authenticated entity will have a unique set of credentials.

This section of the cloud operating model will cover five key areas:

- Hybrid security
- Encryption as a service
- Secrets management
- Advance data protection
- Vault and Azure integrations

Hybrid Security

Dynamic cloud infrastructure means a shift from host-based identity to application-based identity, with low- or zero-trust networks across multiple clouds without a clear network perimeter. In the traditional security world, we assumed high trust internal networks, which resulted in a hard shell and soft interior. With the modern "zero trust" approach, we work to harden the inside as well. This requires that applications be explicitly authenticated, authorized to fetch secrets and perform sensitive operations, and tightly audited.

HashiCorp Vault enables teams to securely store and tightly control access to tokens, passwords, certificates, and encryption keys for protecting machines and applications. This provides a comprehensive secrets management solution. Beyond that, Vault helps protect data at rest and data in transit. Vault

exposes a high-level API for cryptography for developers to secure sensitive data without exposing encryption keys.

Vault also can act like a certificate authority, to provide dynamic short-lived certificates to secure communications with SSL/TLS. Lastly, Vault enables a brokering of identity between different platforms, such as Active Directory on premises and other IAM services to allow applications to work across platform boundaries.

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

Encryption as a Service

Using Vault as a basis for encryption-as-a-service solves difficult problems faced by security teams such as certificate and key rotation. Vault enables centralized key management to simplify encrypting data in transit and at rest across clouds and data centers. This helps reduce costs around expensive Hardware Security Modules (HSM) and increases productivity with consistent security workflows and cryptographic standards across the organization.

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

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

Secrets Management: Secure dynamic infrastructure across clouds and environments

The shift from static, on-premise infrastructure to dynamic, multi-provider infrastructure changes the approach to security. Security in static infrastructure relies on dedicated servers, static IP addresses, and a clear network perimeter. In dynamic infrastructure, security is defined by ephemeral applications and servers, trusted sources of user and application identity, and software-based encryption.

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

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

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

Advanced Data Protection

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

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

Vault and Azure-specific integrations

Azure users can leverage all of the aforementioned Vault features to automate their secrets management and retrieval through Azure-specific integrations. Vault can be automatically unsealed using KMS keys from Azure Key Vault. Secondly, MSI credentials can be used to authenticate systems and applications, which resolves the need to distribute initial access credentials. Lastly, Vault can dynamically generate Azure Service Principals and role assignments. This provides users and applications outside of the cloud an easy method for generating flexible time- and permission-bound access into Azure APIs.

More information on HashiCorp Vault and How Microsoft Azure works with the HashiCorp Product Suite can be found at **hashicorp.com/integrations/microsoft?product=vault**.

Connect and Secure with HashiCorp Consul

HashiCorp Consul is a distributed, highly available service networking platform that can run atop infrastructure in Azure, including on-premise environments and in multiple regions around the world. Consul helps organizations automate networking configurations, enable services to discover one another, and automatically provides secure connectivity between these services.

Addressing Service Discovery Challenges

New challenges emerge during the migration to widely distributed software systems and microservices-based architectures. A common question organizations first encounter is, "how do you keep track of all of your deployed services?" For instance, if Service A is an application deployed into the cloud (web service), how does it know how to find Service B (a database)? In the pre-digital transformation world, a developer would file a ticket to an operator, who would then update the routing tables with the IP address of Service B, and enable access for service A. If service B is moved, then the developer will need to file another ticket to update the tables to ensure that Service B is still reachable. In cloud, service deployment is a much faster process. A ticket-based system to manage service relocations can cause major slowdowns in the application lifecycle as well as friction between operators and developers. Consul automates the process and alleviates the tension in workflows. When Consul is deployed onto a modern hybrid cloud platform like Microsoft Azure, operators can automate the service discovery process. Developers provide a service definition with their applications and Consul captures any configuration changes to services deployed in Azure, such as feature flag updates. These changes are then propagated across the datacenter rapidly — avoiding the inconsistent state that could bring distributed systems down.

Service Networking Automation

Leveraging Consul for Service Discovery introduces developers to the benefits of having a centralized registry to track services. While this addresses the visibility challenge of tracking services across environments, another common bottleneck arises from trying to leverage traditional networking middleware for managing larger service deployments. According to a recent report from ZK Research (zkresearch.com/research/10-networking-priorities-for-digital-transformation-2), a majority of enterprises state that common networking tasks, like provisioning new load balancers, can take days, weeks, and sometimes months. As noted above, this limits the benefits of the cloud as the time to deploy slows down with organizations still relying on manual ticketing processes. The teams responsible for these tasks can also benefit from the automation capabilities that Consul provides.

Consul offers the ability to automate networking configurations for these devices, eliminating the need for manual intervention. Instead of a manual, ticket-based process to reconfigure the traditional network middleware every time there is a change in service network location or configurations, Consul provides a publisher/subscriber (PubSub) like service to automate these operations by updating configuration changes of the network devices. Terraform can also be used to enable rapid day zero operations of the resources used when provisioning new infrastructure.

The newly added service instances will automatically "publish" their location information with the service registry. The network infrastructure can subscribe to service changes from the service registry (by using tools like a Consul Template learn.hashicorp.com/consul/developer-configuration/consul-template or native integration). This enables a publish/subscribe style of automation that can handle highly dynamic infrastructure and scale much higher.

Modern Application Networking with Service Mesh

Moving further in the process, as organizations continue to scale into microservices-based or cloud-native applications (like Azure Kubernetes Service), the underlying infrastructure becomes larger and more dynamic. Modular services need to communicate with each other to compose business logic and functionality, leading to an explosion of east-west traffic.

Existing networking approaches with network appliances cannot effectively handle east-west traffic in dynamic settings. They cause a proliferation of expensive network infrastructure, introduce single points of failure all over the system, and add significant operational overhead to IT teams.

Furthermore, application-based networking has created far greater complexity in the requirements of a networking team than has been needed historically. With significant amounts of workloads becoming ephemeral and highly distributed as microservices, the ability to successfully route application traffic across the network without downtime becomes critical to organizations.

A distributed service mesh pushes routing, authorization, and other networking functionalities to the endpoints in the network, rather than imposing them through a central point in the infrastructure. This makes the network topology simpler and easier to manage, it removes the need for expensive central infrastructure within east-west traffic paths, and it makes service-to-service communication much more reliable and scalable because of the network's decentralized nature. Additionally, it removes the dependency for development teams to incorporate routing and authorization rules directly in application code.

Consul provides an API driven control plane that integrates with sidecar proxies alongside each service instance (such as Envoy, HAProxy, and Nginx) that provide the distributed data plane.

The service mesh approach allows critical functionality such as naming, segmentation and authorization, traffic management, and observability to be configured through policies in the central registry and to be enforced by proxies at the endpoint where an individual service is running.

Consul enables a zero trust network model by securing service-to-service communication with automatic mutual TLS encryption and identity-based authorization. Network operation and security teams can define the security policies through intentions with logical services rather than IP addresses. For example, allowing web services to communicate with databases, instead of IP1 to IP2. Proxies will enforce security consistently regardless of how services scale up and down or migrate to other platforms.

The security benefits of a service mesh based approach are several fold. For most organizations, traffic within a network zone (such as production or PCI) is relatively flat. This means a compromise of a single service would allow an attacker to move laterally to other systems in the same zone. Consul enables a much more fine grained authorization of access to avoid this.

Consul can also be integrated with services like Vault for centralized PKI and certificate management.

To address the application networking concerns, layer 7 routing and traffic policy management is provided by Consul and enforced by routing traffic based on many possible conditions (HTTP header, path based routing, etc.) to support use cases such as canary, A/B testing and gradual application deployment rollouts, and application lifecycle efforts. These practices have become the foundation of progressive delivery for applications in the enterprise and can only be effectively achieved leveraging a service mesh. For cross-cloud communications, Consul's Mesh Gateway feature routes traffic to the correct endpoint on a private network without requiring expensive IPSec or MPLS connectivity.

Consul, containers, and AKS

Consul can also successfully extend modern container management platforms such as Azure Kubernetes Services (AKS) and Azure Service Fabric. While both Kubernetes and Service Fabric provide their own service discovery and health checking mechanisms, Consul allows those platforms to integrate with services that reside outside of their management boundary. For example, a web service or a database running outside of the Kubernetes cluster, and even potentially an on-prem data center, can be configured to be discoverable by services deployed on Kubernetes via Consul.

Empowering customers together

"The visibility, transparency, and control we have with Consul eliminates so many of the service discovery and connectivity obstacles that used to prevent us from working as quickly and efficiently as we wanted... Consul lets us spread more than 200 microservices over several AKS clusters. Each AKS cluster connects to a local Consul client, which feeds into a Consul cluster that forms a larger service discovery mesh that allows us to find and connect services in a matter of minutes with minimal effort."

-Sriram Govindarajan, Principal Infrastructure Engineer, Mercedes-Benz Research & Development (MBRDNA)

Using HashiCorp Cloud Service (HCS) on Azure as a Managed Service

For customers who would like an easier way to get critical applications running quickly in Azure based environments and adopt Consul to enable service-to-service communications, HashiCorp provides a managed service offering to offload teams the burden of managing a distributed runtime system. Since HCS on Azure is a fully managed service, HashiCorp will handle all of the operational tasks including provisioning, monitoring, troubleshooting, and server upgrades on behalf of the customer. This allows the customer to focus on application and workload-specific concerns that are of highest interest to the business.

HCS is offered as an Azure managed application. In addition to alleviating the operational maintenance burden, HCS is integrated into the user dashboard for a native Azure experience. Organizations can securely deploy customers using a push-button experience in the Azure portal, pay for the service using integrated Azure billing, and integrate identity management with Azure Active Directory (Azure AD). As an Azure service, HCS supports both discovery and service mesh. As a service mesh, HCS enables automated connections between Azure services and encrypts the traffic using mTLS. You can even deploy AKS and Consul using the same HashiCorp Terraform template.

HCS is preconfigured for production workloads across three key deployment scenarios: Azure Kubernetes (AKS), Azure VMs, and Hybrid Kubernetes/VMs. HCS on Azure is the easiest way to enable Service Mesh and secure service-to-service connectivity for Kubernetes workloads on Azure. It also enables both Service Discovery and Service Mesh across multiple Kubernetes and VM environments.

The benefits of a single control plane

- Consul provides the control plane for multi and hybrid-cloud networking
- Centrally control the distributed data plane to provide a scalable and reliable service mesh
- Automate centralized network middleware configuration to avoid human intervention
- Provide a real-time directory of all running services to improve application inventory management
- Enable visibility into services and their health status to enhance health and performance monitoring
- Automate lifecycle management of certificates which can be issued by 3rd party Certificate Authority
- Provide unified support across a heterogeneous environment with different workload types and runtime platforms

Leveraging HashiCorp Consul with Azure

- Enable services running in any Azure region or on-premise environment to discover one another quickly and efficiently
- Reduce deployment time of applications using Consul's dynamic load balancing features with existing middleware (like F5, NGINX, or HAProxy)
- Enhance the Kubernetes experience by leveraging AKS and Consul's service mesh capabilities

Hybrid-Cloud Application Delivery

Finally, at the application layer, new apps are increasingly distributed while legacy apps also need to be managed with more flexibility. HashiCorp Nomad provides a flexible orchestrator to deploy and manage legacy and modern applications, for all types of workloads: from long running services, to short lived batch, to system agents. To achieve shared services for application delivery, IT teams should use Nomad in concert with Terraform, Vault, and Consul to enable the consistent delivery of applications on cloud infrastructure, incorporating necessary compliance, security, and networking requirements, as well as workload orchestration and scheduling.

Mixed Workload Orchestration

Many new workloads are packaged in containers and developed with the intent to deploy Kubernetes or other container management platforms. However, most legacy workloads will not be moved onto those platforms, nor will future serverless applications. As a result, organizations end up with the need to orchestrate mixed workloads. Nomad can handle the deployment of all workloads from virtual machines, through standalone binaries, to containers, and provides core orchestration benefits across those workloads such as release automation, multiple upgrade strategies, bin packing, and resilience. Platforms such as Kubernetes only handle container workloads and also require specialist skills to operate due to the complex nature of the platforms. In contrast, Nomad is focused on simplifying orchestration and delivering highly efficient scheduling. Nomad provides a consistent workflow at scale in any environment.

High Performance Compute

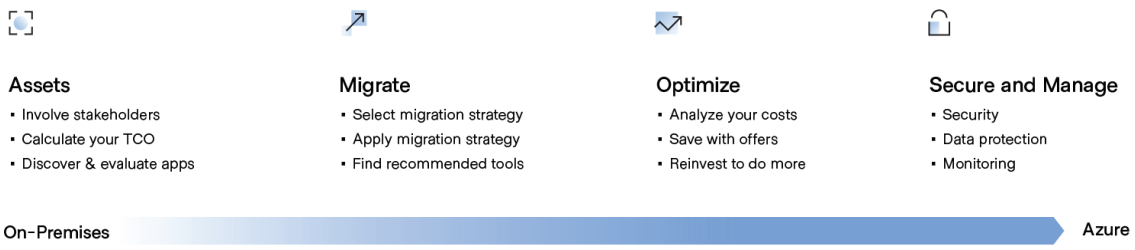
Nomad is designed to schedule applications with low latency across very large clusters. This is critical for customers with large batch jobs, as is common with High Performance Computing (HPC) workloads. In the million container challenge, Nomad was able to schedule one million instances of Redis across 5,000 machines in three data centers, in under 5 minutes. Several large Nomad deployments run at even larger scales.

Multi-Datacenter Workload Orchestration

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

In conclusion

The transition to cloud, and hybrid-cloud, environments is a generational transition for IT. For most enterprises, this transition means shifting from a static to dynamic environment and focusing on developing your Azure migration plan around people, process, and tools to drive the right outcomes for your organization.



Developing a cloud migration plan

1. **Create a cloud migration plan**
   By establishing your cloud migration priorities and objectives before you start planning, you can ensure a more successful migration. Automated cloud migration tools will also provide insights into your environment and dependencies to build out your cloud migration project plans.

2. **Involve stakeholders**
   Reach out to key people throughout the organization — this should include representation from both IT and the involved business owners. Getting everyone's engagement and support before you migrate will lead to a smoother, faster cloud migration process that meets everyone's goals.

3. **Calculate your Total Cost of Ownership**
   Evaluate the potential cost savings of migrating to Azure by calculating and comparing your total cost of ownership (TCO) for Azure with that of a comparable on-premises deployment. HashiCorp has developed documentation to help organizations better estimate their Azure cloud costs and can be found at terraform.io/docs/cloud/cost-estimation/azure.html.

4. **Discover and Evaluate Apps**
   To start any migration, compile an inventory of the physical and virtual servers in your environment. While your current management tools may provide a good representation of the hundreds — maybe thousands — of applications your organization is running, you need an inventory mechanism that can feed data into subsequent steps.

Aligning people, process, and tools to the journey

- **People: Shifting to hybrid-cloud skills**
    - Reuse skills from internal data center management and single cloud management.
    - Embrace DevSecOps and agile practices to continuously deliver increasingly ephemeral and distributed systems.
- **Process: Shifting to self-service IT**
    - Position central IT as enabling shared services focused on application delivery velocity — in order for the organization to ship software more rapidly with minimal risk.
    - Establish centers of excellence across each layer of the cloud for self-service delivery.
- **Tools: Shifting to dynamic environments**
    - Workflows, not technologies — use tools that support the increasing ephemerality and distribution of infrastructure and applications that support critical workflows rather than being tied to any specific technology.
    - Maintain compliance and manage risk in a self-service environment through providing policy and governance tooling.

About HashiCorp & Microsoft. Better together.

HashiCorp and Microsoft are longstanding partners in the cloud infrastructure community. In 2017, Microsoft committed to a multi-year partnership aimed at further integrating Azure services with HashiCorp products. As a result of this collaboration, organizations can rely on tools like Terraform to create and manage Azure infrastructure. The tight integration and support for Azure allows operators to easily deploy resources on Azure using Terraform and secure them via Vault. Additionally, Microsoft utilizes HashiCorp tools for internal use. Packer is trusted for the creation of new Linux images for Azure services. This collaboration enables Microsoft and HashiCorp to create new and innovative ways for their products to integrate further, easing the cloud adoption journey for enterprise organizations.


HashiCorp