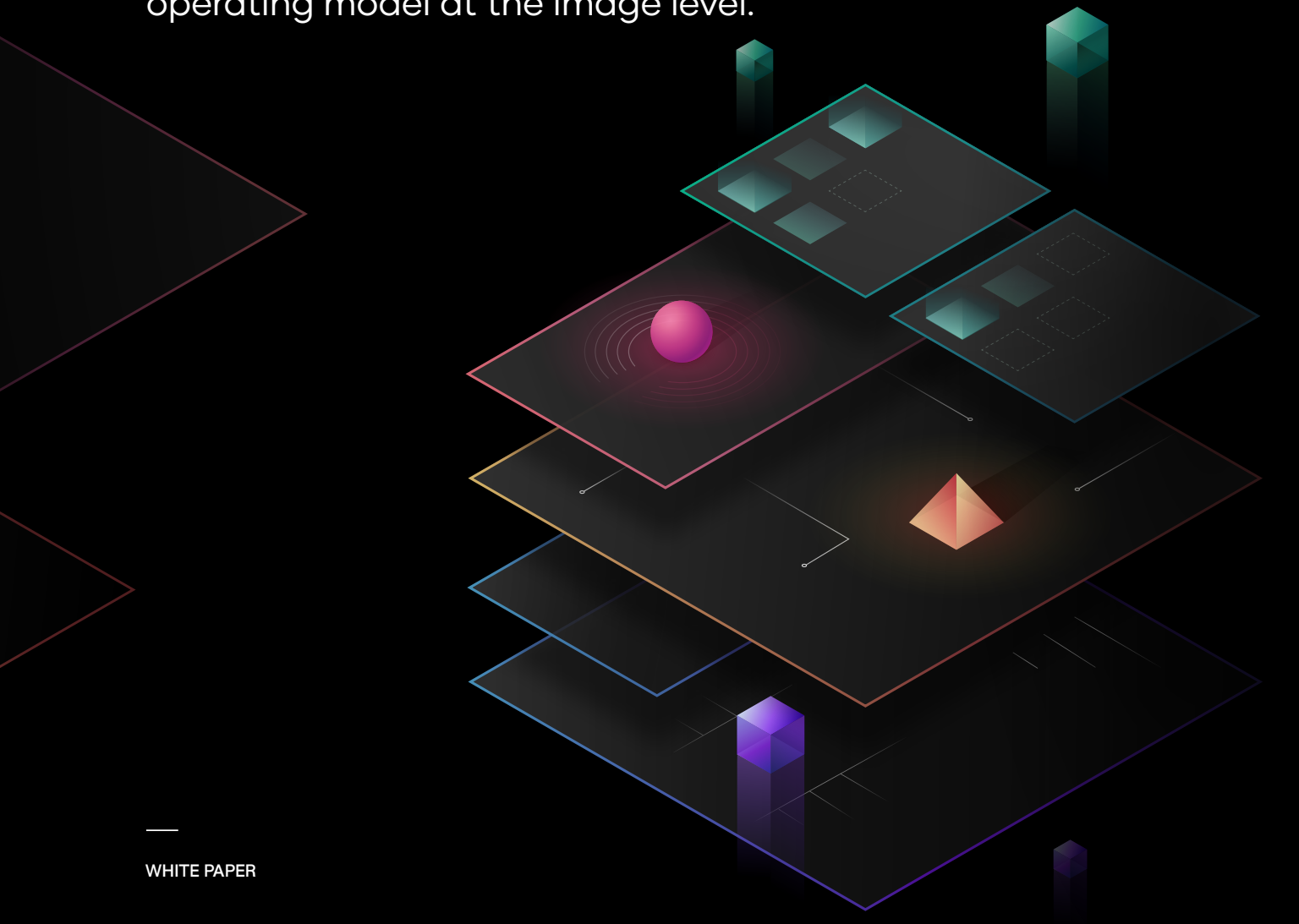
HashiCorp

Unlocking your cloud operating model: Image management

Understand the benefits of adopting a cloud operating model at the image level.



WHITE PAPER

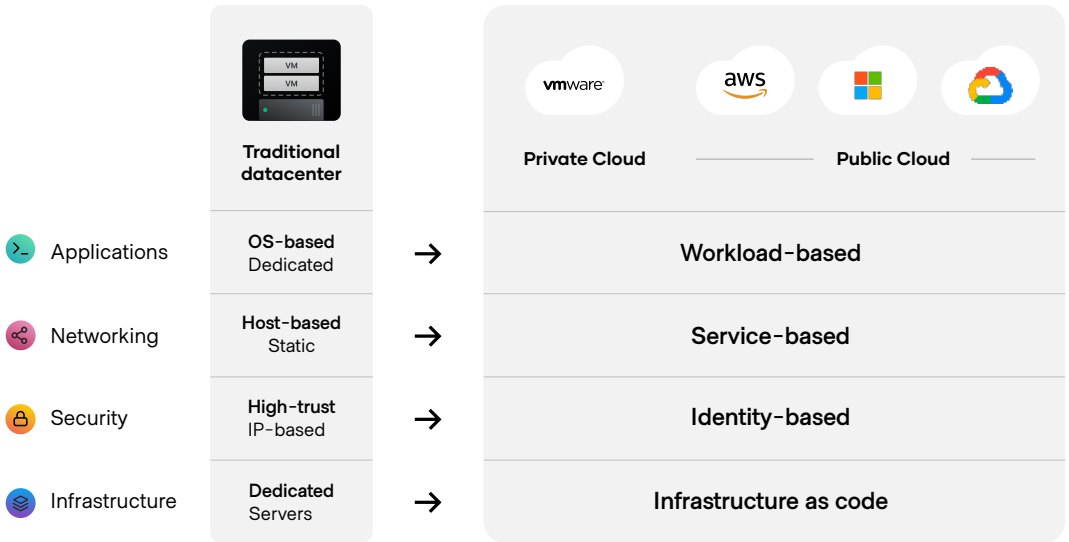
Overview

This white paper looks at the implications of the cloud operating model and describes its benefits for standardizing image creation and management. This is part of a series of white papers on infrastructure as code (IaC) for provisioning, cloud compliance and management, and minimizing cloud waste.

Adopting a cloud operating model at the image level enables organizations to standardize image creation, track all associated builds, automate provisioning pipelines, and simplify image lifecycle management. Implementing these practices lowers the risk of deploying insecure images, reduces deployment times, and improves efficiency through automated image management.

Adopting a cloud operating model

Moving to the cloud involves a shift from static to dynamic infrastructure. This transformation focuses on moving away from manually configuring and managing a fixed set of IT resources and instead focusing on running dynamic resources on demand in automated workflows. For most organizations, the goal of this transition is to enable innovation, delivering new business and customer value faster and at a larger scale.



As teams move from traditional datacenters to the cloud, the foundations of each layer change.

To fully realize the benefits of cloud computing, organizations must transition to scalable dynamic workflows and management at each cloud layer. From here they can establish central shared services enabled by platform teams to improve speed, increase efficiency, and reduce risk. HashiCorp Terraform serves as the industry standard for multi-cloud provisioning and enables organizations to utilize shared services at the infrastructure layer, but within this layer sit the building blocks of modern infrastructure, system images.

Multi-cloud image challenges

Images (such as AMIs for Amazon EC2, virtual machines, Docker containers, and more) are the building blocks of modern computing infrastructure. Organizations adopting multi-cloud typically start by using Terraform for centralized provisioning, but Terraform does not handle the details of image creation and management. For that, you can modify infrastructure in place with configuration management tools, or you can take the approach that is considered more stable across the IT world - immutable infrastructure. Immutable infrastructure remains untouched following deployment; instead of being modified, it is destroyed and replaced with a fresh iteration during each infrastructure update. This approach helps ensure consistent, reliable, and secure deployments better suited to support the demands of multi-cloud environments.

As organizations deploy fleets of images to support services across cloud and private environments, the complexity and scope of these services often involve multiple different teams. Without consistent, central processes and tooling in place, organizations can experience variability in their imaging workflows creating several challenges:

- **Inconsistencies:** With different image practices, teams are prone to achieve different outcomes with varying levels of infrastructure performance.
- **Risks:** Manual, checklist-driven procedures to apply security standards lead to human error in the form of misconfigured and insecure images that can introduce security threats to the organization and result in outages.
- **Delays:** Teams may duplicate efforts and spend excessive time manually building images for different environments, increasing time to deployment.

To combat these issues, organizations and their platform teams need to establish a central shared service for their image creation and management workflows, with consistent processes in place to:

- **Establish standardization:** Synchronize tooling and approved templates across teams to minimize edge cases that may undermine the overall health of the final infrastructure delivered.
- **Enforce security:** Standardize security and compliance testing to confirm adherence to guardrails defined by the organization before images are released for consumption.
- **Improve efficiency:** Deploy faster by automating image provisioning and version updates across downstream provisioning pipelines.

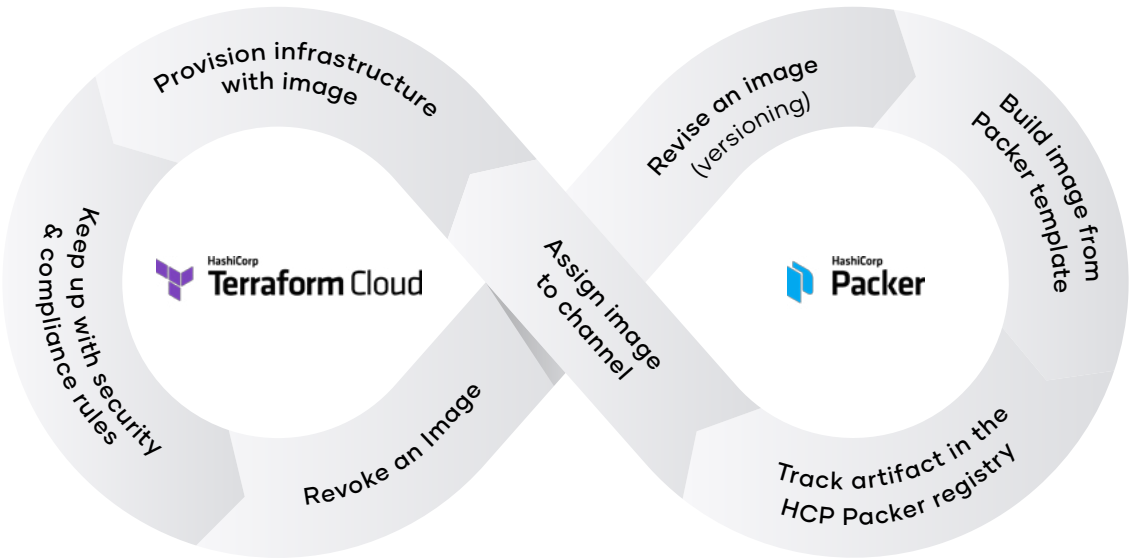
Solution: Integrating HCP Packer into Terraform Cloud

Using Terraform Cloud with HCP Packer allows platform teams to unify their image management workflows with their provisioning processes. This integration enables users to shift security and governance left to the image level and create a golden image pipeline to automate image management across downstream builds and provisioning pipelines.

HCP Packer helps platform teams establish a unified image management system across groups within an organization. This provides embedded policy and governance, organization-wide visibility, ease of integration with peripheral technologies, and overall reliability at scale.

By integrating HCP Packer into their multi-cloud workflows, organizations can:

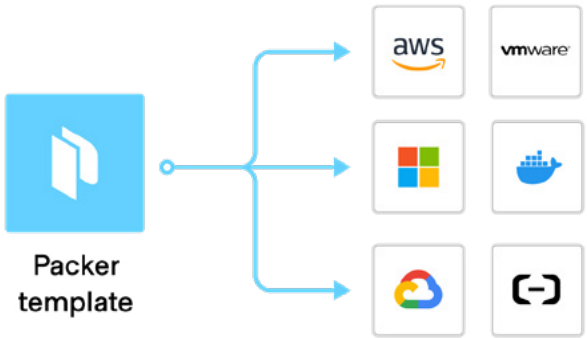
- Standardize image creation to ensure all builds deployed are secure and compliant.
- Track all image builds and associated metadata in a central artifact registry.
- Automate provisioning pipelines and continuously monitor infrastructure health.
- Simplify image lifecycle management.



Golden image pipeline with Terraform Cloud and HCP Packer

Standardize image creation across clouds

HCP Packer is a managed extension of HashiCorp Packer, a free, source-available tool that has become an industry standard for creating identical image builds for multiple cloud and on-premises platforms from a single source configuration file. Packer is lightweight, runs on every major operating system, and is highly performant, enabling multiple image builds to be created in parallel. This standard workflow for generating builds across your multi-cloud environment ensures imaging processes are consistent and repeatable regardless of the type of artifact you are building. Examples include AMIs, Azure VM images, VM templates for VMware vSphere, Docker containers, or Vagrant boxes.



Create identical images for multiple platforms from a single source configuration.

The first step in creating a golden image pipeline is to create a set of golden images with Packer. A "golden image" is an approved image that acts as a template on top of which developers can build applications. Quality control teams in an organization will ensure that these images have the most up-to-date common system packages, logging and monitoring tools, security patches, and configuration hardening.

Manual configuration and patching can lead to inconsistent outcomes and increased risk of security breaches and compliance violations due to unsecured or out-of-date base images. To prevent this, Packer helps teams coordinate common requirements and implement them uniformly across multiple operating platforms via golden images. Codifying these organizational requirements helps ensure all images are consistent, secure, and compliant before deployment. Golden image versions can then be easily updated and released to react to emerging business, technology, or security conditions.

Packer simplifies golden image creation by enabling organizations to leverage the HashiCorp Configuration Language (HCL). This simple syntax and human-readable language lets users define and describe images using a declarative approach, defining an intended end-state rather than the individual steps to reach the desired outcome. HCL simplifies the process of embedding all organizational requirements —such as security and operational details — into golden images. Codification also enables collaboration; changes can be reviewed by the appropriate stakeholders using standard version-control workflows before being implemented.

Packer also provides an extensive template library that enables users to leverage common configurations across multiple image builds. Templates consist of a series of declarations and commands for Packer to follow when generating a new image build. The template specifies what plugins (builders, data sources, provisioners, post-processors) to use, how to configure each of those plugins, and in what order to run them.

Track and govern images at scale

When a new golden image is created, this new version is automatically published to HCP Packer. HCP Packer serves as a managed registry that stores image metadata, including when they were created, the associated cloud provider, and any custom labels specified in your image build. The HCP Packer artifact registry helps you track information about images, clearly designate which versions are approved for consumption, and query the right images to use in both Packer and Terraform configurations. Access to this centralized library helps align the workflows of image creation and deployment, allowing operations and development teams to work together to manage, track, and govern all artifacts across your infrastructure estate.

Manage images with channels

Image channels is a core feature of HCP Packer that enables collaboration across teams. With channels, you can label image versions, known as iterations, to describe the quality and stability of a build. By assigning human-readable names to image iterations, downstream consumers can easily reference the images in Packer templates and Terraform configurations.

For example, you can designate a specific channel for testing, allowing users to promote new versions and quickly spin up an instance to validate the image. Once the new version passes the required tests, it can be promoted to the stable channel, alerting downstream consumers that it is approved and ready for deployment. This workflow provides teams with vetted, ready-built artifacts that supply standard services in a plug-and-play fashion. Consumers can tailor versions of artifacts to streamline their efforts in the updating and release stages, and ensure they are referencing the latest version without having to update their code directly.

As new image versions are published, assigned, and revoked over time, it is important to maintain visibility into their history and downstream dependencies. HCP Packer's image ancestry tracking gives users visibility into relationships between the new image iteration (child) and its source iteration (parent), if any.

Channel assignment history provides a complete record of artifact activity in a channel. You can browse any existing bucket and select a channel to see exactly which iterations have been made available to downstream consumers. From there you can view each image iteration's channel history, the status of its parent image, and extended metadata. You can also view when the iteration was assigned and by whom.

Image builders need to collaborate with other stakeholders to validate that new image iterations meet compliance and functionality requirements before releasing them to downstream consumers. Restricted channels offer control over the release of images by providing a means to limit channel access for other collaborators. This granular permissions control lets you ensure only the necessary users have channel access and enables the least privilege principle. This addition also helps streamline the image-validation process and prevents downstream consumers from using new image iterations before they have been validated and approved.

Automate provisioning and monitor infrastructure health

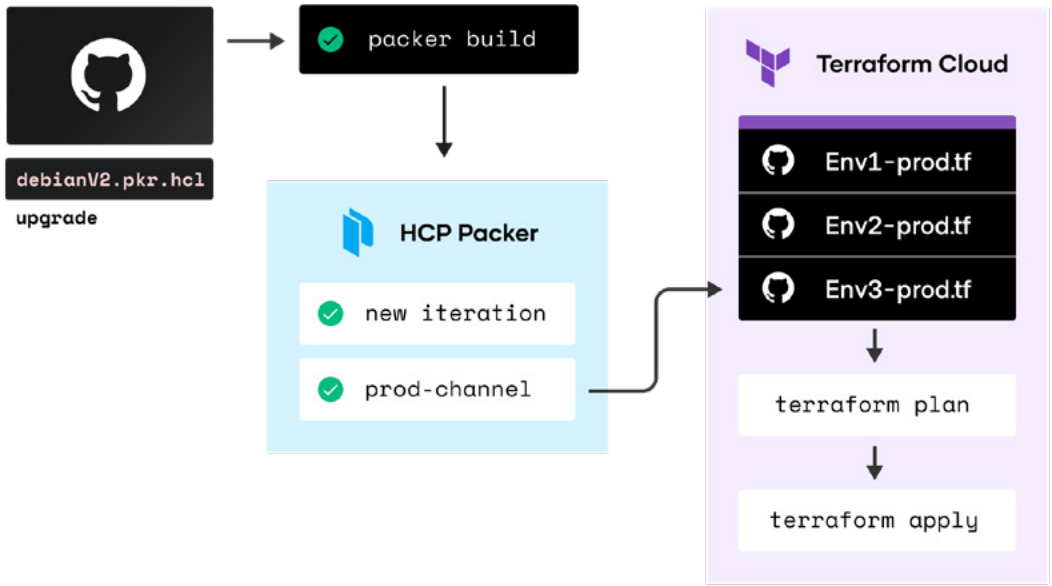
Organizations invest heavily in developing, implementing, and supporting workflows that help build better applications, deploy more effectively, and continually improve their offerings. In early-stage cloud adoption, provisioning workflows tend to vary depending on the target cloud platforms or private environments. Organizations quickly begin to see the need for uniform instrumentation that fosters a common provisioning practice.

One reason Terraform has become the industry standard for solving this problem is its environment-agnostic approach to provisioning resources. A unified approach for image deployment across on-premises and different cloud vendors is also crucial. This unification can be achieved by integrating HCP Packer into existing Terraform Cloud workflows. Teams that consume images in Terraform can rely on approved, standard subprocesses in the initial provisioning stages, and continue to evaluate whether iterations uphold security and compliance standards over time.

With a golden image built, published, validated, and promoted to your organization's stable channel, Terraform runs referencing the updated version can be queued automatically for any workspace using the channel. The image updates across downstream provisioning pipelines can take place autonomously with auto-apply settings or be gated by manual approval processes. The Terraform Cloud run task for HCP Packer helps prevent the deployment of non-approved images with:

1. **Data source image validation** to scan your Terraform plan for references to the HCP Packer iteration and image data sources, warning you or blocking the run if any referenced data is associated with a revoked image version
2. **Resource image validation** to scan your Terraform configuration for resources that use hard-coded machine image IDs and check if the image is tracked by HCP Packer. It also warns users if the image is associated with a revoked iteration and prompt them to reference the HCP Packer data source instead for better tracking and management capabilities

With this automation, teams can integrate images easily onto a larger workflow framework to complement automated delivery pipelines.



Reference HCP Packer in your Terraform Cloud workflows.

Drift detection and continuous validation

Once the new image version is successfully approved and provisioned, the next step is to perform health assessments to make sure this infrastructure doesn't change over time. Even with a standardized initial provisioning process, settings on infrastructure can still be modified or circumvented, opening up your infrastructure to the possibility of configuration drift. Drift is the term for when the real-world state of your infrastructure differs from the state defined in your configuration. Drift occurs when a user modifies resources outside of the Terraform workflow.

For example, a colleague may update resource configurations directly in the cloud provider console to resolve a production incident. Terraform Cloud's drift detection allows users to actively monitor their

infrastructure for these changes and receive alerts when they take place. From the health assessments dashboard they can quickly uncover the root cause for the change, determine if it is necessary, and accept it or automatically remediate if not.

Terraform Cloud can also perform health checks for custom conditions and assertions with continuous validation. Users can monitor whether the functional validations defined in Terraform code continue to pass over time and receive an alert when an assertion fails. For example, you can monitor whether your website returns an expected status code, whether an API gateway certificate is valid, or whether the image artifact referenced from an HCP Packer channel is too old or has a scheduled revocation. Identifying failed checks helps you proactively resolve failures and prevent errors during your next Terraform operation.

These two features provide users with flexible options to validate their infrastructure uptime, health, and security — all in one place without requiring additional tools.

Manage image lifecycles

If one of your golden images is outdated or possesses a vulnerability, you may need to revoke it to prevent infrastructure deployments from using it. HCP Packer and Terraform Cloud help provide a fast, unified, and simple revocation workflow across downstream builds and provisioning pipelines. When a golden image version is updated in an HCP Packer channel, any deployments using that image are simply re-run to pick up the new association. The technical effort to support the update is transparent for the practitioner because the Terraform data source integrates directly with HCP Packer to query the channel without hard-coding image identifiers. HCP Packer offers this simplified revocation workflow in three ways:

1. **Scheduled revocation:** Plan a revocation for a future end-of-life (EOL) date or revoke the image version immediately.
2. **Inherited revocation:** Building on HCP Packer's image ancestry tracking and established parent/child relationship to revoke just the base golden image or all associated downstream dependencies.
3. **Channel rollback:** Building on channel assignment history to provide quicker remediation of released artifacts by providing the option to roll back channels to their previously assigned iteration. This also works with HCP Packer's inherited revocation to automatically roll back the channel assignments of any descendant images when a parent image is revoked.

This simplified revocation workflow allows organizations to reduce their time to remediation during a security incident without impacting downstream provisioning processes.

Business impact

Integrating HCP Packer's image management capabilities into existing Terraform Cloud workflows brings a number of key benefits:

Lower risk

Never deploy insecure images: Embed security and compliance requirements into all images across your cloud environments, set EOL dates, and automate revocation.

Faster speed

Decrease time to deployment: Speed deployment by creating and reusing images from a single-source configuration file, connecting to VCS, and collaborating across teams,

Increased efficiency

Automate image management: Standardize image versions, change a golden image once, and automatically update across downstream builds.

Summary and next steps

Extend a cloud operating model to the image level to ensure consistent and secure infrastructure workflow across your organization as it grows in scale.

Try Terraform Cloud and HCP Packer for free to begin unifying your imaging and provisioning workflows and simplifying infrastructure lifecycle management.

Link your Terraform Cloud and HashiCorp Cloud Platform (HCP) accounts together for a seamless sign-in experience.

HashiCorp