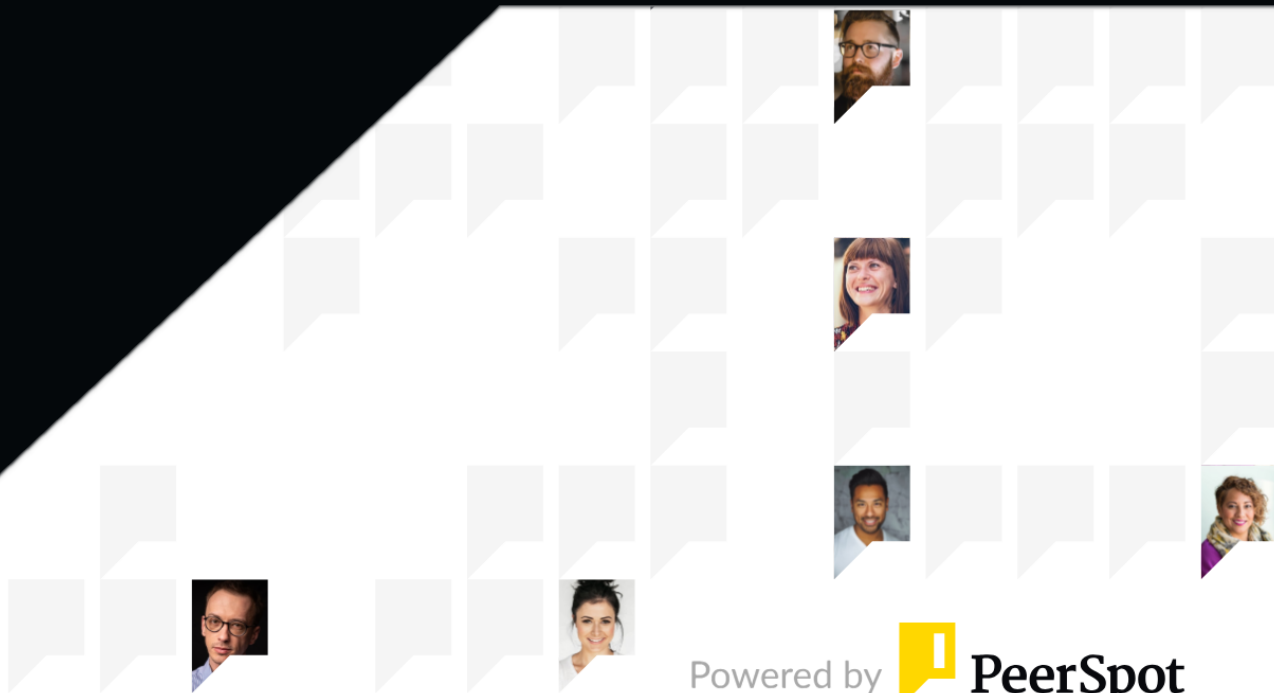




**IBM HashiCorp Infrastructure Lifecycle
Management**

**Reviews, tips, and
advice from real users**



Powered by  **PeerSpot**

Contents

Product Recap..... 3 - 4

Valuable Features..... 5 - 8

Other Solutions Considered..... 9 - 10

ROI..... 11

Use Case..... 12 - 14

Setup..... 15

Customer Service and Support..... 16

Other Advice..... 17 - 19

About PeerSpot..... 20 - 21

Product Recap



IBM HashiCorp Infrastructure Lifecycle Management

IBM HashiCorp Infrastructure Lifecycle Management Recap

IBM HashiCorp Infrastructure Lifecycle Management offers a comprehensive approach to automating infrastructure provisioning and management, tailored for enterprises seeking streamlined operations.

IBM HashiCorp Infrastructure Lifecycle Management leverages automation to simplify the provisioning, configuration, and management of cloud infrastructure. It integrates with existing workflows to optimize resource allocation and reduce manual intervention. This alignment with both private and public cloud environments ensures scalability and flexibility for businesses looking to enhance their cloud strategies.

What are the key features of IBM HashiCorp Infrastructure Lifecycle Management?

- **Infrastructure as Code:** Facilitates the automation of infrastructure provisioning using code, enhancing consistency and repeatability.
- **Multi-Cloud Support:** Offers a unified interface to manage infrastructure across different cloud vendors.
- **Compliance Automation:** Ensures infrastructure security and compliance by automating policy enforcement.

What benefits or ROI can users expect?

- **Reduced Operational Costs:** Automation decreases the need for manual intervention, lowering management expenses.
- **Improved Resource Utilization:** Optimizes allocation of resources, leading to increased efficiency and performance.
- **Enhanced Agility:** Accelerates deployment processes, allowing faster response to business changes.

Industries such as finance and healthcare implement IBM HashiCorp Infrastructure Lifecycle Management to ensure secure, compliant, and efficient cloud strategies. Its adaptability to industry-specific regulations makes it a valuable choice for companies needing reliable infrastructure management.

Valuable Features

Excerpts from real customer reviews on PeerSpot:



“Overall, IBM HashiCorp Infrastructure Lifecycle Management has been a practical solution for standardizing infrastructure management and reducing manual provisioning work.”



Verified user

Developer at a healthcare company with 1,001-5,000 employees



“IBM HashiCorp Infrastructure Lifecycle Management has positively impacted my organization by significantly reducing manual infrastructure work.”



PrinceKumar7

Cloud Engineer at a computer software company with 501-1,000 employees



“Before using Terraform, our team used manual setup to create clusters in AWS using the AWS console, which would extend into two to three sprints for one person to set up all those components, and with Terraform and those reusable modules, we were able to minimize it to two to three days with all verifications included.”



Gardihewagesathira Gardihewa

DevOps Engineer at a consumer goods company with 10,001+ employees

What users had to say about valuable features:

“The best features IBM HashiCorp Infrastructure Lifecycle Management offers include having a central place with reusable templates. Since this is a multi-cluster setup, coordinating a shared change safely is essential. Most of the time, there are infrastructure patterns in a platform, so we need a repeatable way to use them across environments.

“Since we manage a separate Terraform modules repository for our configurations with AWS EKS cluster configurations and HashiCorp Vault configurations, if we need to make a change across all clusters, we only need to change that Terraform module itself.

“Another benefit is that it can be used not only for a specific cloud setup. We can use it for AWS, GCP, and also for Azure. The multi-cluster, multi-cloud setup capability using Terraform is a good feature..”

Gardihewagesathira Gardihewa

DevOps Engineer at a consumer goods company with 10,001+ employees

[Read full review](#) 

“The best features IBM HashiCorp Infrastructure Lifecycle Management offers include standardization, which I find most valuable. Terraform is powerful, but the real operational value comes from having a consistent lifecycle around the infrastructure. Instead of trying to remember how someone created an AWS resource six months ago, I can look at the infrastructure definition and understand how it is supposed to be managed. Other major benefits include repeatability; if we need another environment, we do not have to start from scratch.

“Standardization and repeatability have greatly helped my team and projects. Rather than having every application team define the VPC differently, we can standardize it and use that standardization everywhere, leading to a more consistent infrastructure. This consistency makes it much easier to provision the infrastructure each time and saves us a significant amount of time.

“IBM HashiCorp Infrastructure Lifecycle Management has positively impacted my organization by significantly reducing manual infrastructure work. Before standardizing our infrastructure as code workflows, creating an environment involved extensive manual efforts, AWS console work, and documentation. With reusable Terraform configurations, much of that process becomes repeatable and results in faster environment creation. Once the required modules and variables are available, setting up another environment is much easier. Consistency has improved, troubleshooting is simpler, change control is better, and disaster recovery is easier now..”

PrinceKumar7

Cloud Engineer at a computer software company with 501-1,000 employees

[Read full review](#) 

“The best features I found most useful in IBM HashiCorp Infrastructure Lifecycle Management are reusable Terraform modules, infrastructure as code with version control, and automated provisioning across environments. I also appreciate the plan and apply workflow because we can review infrastructure changes before deployment and maintain a clear history of what changed.

“I rely most on the infrastructure as code feature because it gives us a consistent way to manage infrastructure through version-controlled configuration. It also makes changes easier to preview, reproduce, and roll back compared with making changes manually in the cloud console.

“One thing I wish I had understood earlier regarding the features is the importance of designing Terraform modules and state management properly from the beginning. I also find the integration with CI/CD and policy controls valuable because it helps us enforce standards without adding too much manual work.

“IBM HashiCorp Infrastructure Lifecycle Management has mainly improved consistency and reduced the time needed to provision and update infrastructure. We have also seen fewer configuration issues because changes go through a standardized version control process instead of being handled manually..”

Verified user

[Read full review](#) 

Developer at a healthcare company with 1,001-5,000 employees

Other Solutions Considered

“Before choosing IBM HashiCorp Infrastructure Lifecycle Management, I evaluated other options including self-managed Terraform workflows and internal Terraform modules..”

PrinceKumar7

Cloud Engineer at a computer software company with 501-1,000 employees

[Read full review](#) 

“Before IBM HashiCorp Infrastructure Lifecycle Management, I relied more on manual cloud console provisioning and some custom scripts for infrastructure setup. I moved to HashiCorp because I needed a more standardized, version-controlled approach that could scale across multiple environments..”

Verified user

Developer at a healthcare company with 1,001-5,000 employees

[Read full review](#) 

“I looked at a few alternatives before choosing IBM HashiCorp Infrastructure Lifecycle Management, mainly AWS CloudFormation, Azure Resource Manager, and compared them around multi-cloud support, ease of managing infrastructure as code, reusable components, teamwork flow, and integration with our existing CI/CD processes..”

Verified user

Developer at a healthcare company with 1,001-5,000 employees

[Read full review](#) 

ROI

Real user quotes about their ROI:

“I have seen a return on investment, particularly in terms of time saved, as previously mentioned regarding reusability and standardization; time saved is a key metric..”

PrinceKumar7

Cloud Engineer at a computer software company with 501-1,000 employees

[Read full review](#) 

“I have seen a positive ROI mainly through time savings rather than reducing headcount. As a rough estimate, infrastructure provisioning and routine changes take thirty to forty percent less effort, which frees up teams to focus more on higher-value engineering work..”

Verified user

Developer at a healthcare company with 1,001-5,000 employees

[Read full review](#) 

Use Case

“My main use case for IBM HashiCorp Infrastructure Lifecycle Management is infrastructure provisioning and lifecycle management using Terraform. Day-to-day, I use it to provision cloud resources through reusable modules, manage infrastructure changes through version-controlled code, and keep deployments consistent across development, testing, and production environments.

“One thing I would add about our main use case is that it gives us better control and visibility over infrastructure changes. It fits well into our CI/CD process, so provisioning and updates are more repeatable, while approvals and version history help reduce the risk of manual changes..”

Verified user

Developer at a healthcare company with 1,001-5,000 employees

[Read full review](#) 

“I am working in a platform team where we manage an internal developer platform with lots of workloads around AWS EKS and GCP GKE. We use Terraform to provision infrastructure across those cloud setups. Terraform allows us to create infrastructure as code to have a central place where all configurations are stored with all infrastructure setups.

“Recently, we had a requirement from a customer, which is a developer team. They needed to deploy an application in a specific, dedicated AWS EKS cluster. We had to create that cluster and configure the networking and all cluster bootstrap modules that we needed to set up. We provisioned the AWS EKS cluster using Terraform and our reusable Terraform modules that we use to provision our infrastructure across the cloud environment.

“Since we have specific Terraform modules for each infrastructure setup, we can reuse those modules in our infrastructure repository and provision a cluster within a few minutes rather than rewriting all infrastructure configurations. This is particularly helpful for us as a platform team managing an internal developer platform with many customers..”

Gardihegathira Gardihewa

DevOps Engineer at a consumer goods company with 10,001+ employees

[Read full review](#) 

“Our main use case for IBM HashiCorp Infrastructure Lifecycle Management is standardizing and automating the lifecycle of cloud infrastructure using IaC, mainly for AWS environments. We have multiple environments and a number of AWS services that need to be provisioned consistently. Instead of having our engineers manually create resources through the AWS console, we use a Terraform-based infrastructure definition so that the infrastructure can be reviewed and version controlled and reproduced anytime. The important part is not simply creating an EC2 instance or VPC with Terraform, but having a consistent process for building, changing, reviewing, and eventually removing infrastructure. IBM HashiCorp Infrastructure Lifecycle Management positions infrastructure lifecycle management around a standardized workflow for provisioning, managing, and governing infrastructure with Terraform at the center of this workflow.

“A quick, specific example of how I have used it in a real situation involves our IaC workflows, mainly Terraform, for both new infrastructure changes and existing infrastructure. A typical workflow includes Terraform configuration, then review, plan, approval, and application, resulting in the AWS infrastructure being up. This provides us with a much more controlled process than manually making infrastructure changes. For example, if we need to create a new application environment, we can reuse the same infrastructure patterns while changing environment-specific variables, which helps avoid situations where development, staging, and production become configured differently. We maintain a reusable Terraform configuration or module for common components, allowing us to standardize VPC definitions rather than having every application team define a VPC differently, which significantly simplifies the infrastructure review process. One practical benefit is that infrastructure changes become visible before they are applied, giving us an opportunity to review what is going to change before applying it, particularly useful in production..”

PrinceKumar7

Cloud Engineer at a computer software company with 501-1,000 employees

[Read full review](#) 

Setup

The setup process involves configuring and preparing the product or service for use, which may include tasks such as installation, account creation, initial configuration, and troubleshooting any issues that may arise. Below you can find real user quotes about the setup process.

“The pricing for IBM HashiCorp Infrastructure Lifecycle Management was reasonable for the value I received, although the overall cost depends on the deployment model, usage, and the specific HashiCorp capabilities licensed. Setup was fairly manageable, but I did spend some initial effort on architecture, integrations, and establishing Terraform standards..”

Verified user

[Read full review](#) 

Developer at a healthcare company with 1,001-5,000 employees

Customer Service and Support

“Customer support for IBM HashiCorp Infrastructure Lifecycle Management has been generally good, especially for product-specific or technical issues. For more complex Terraform or integration problems, I sometimes rely on documentation and community resources as well, since those can provide more detailed troubleshooting examples.

“I would rate the customer support for IBM HashiCorp Infrastructure Lifecycle Management eight out of ten. The response has generally been helpful for product-related issues, although more detailed guidance for complex integrations and troubleshooting would make it stronger..”

Verified user

Developer at a healthcare company with 1,001-5,000 employees

[Read full review](#) 

Other Advice

“I would advise others looking into using IBM HashiCorp Infrastructure Lifecycle Management that if you have a complex infrastructure where maintainability and standardization become valuable, this is a good tool to use. I would rate this product a 9 out of 10..”

PrinceKumar7

Cloud Engineer at a computer software company with 501-1,000 employees

[Read full review](#) 

“I would recommend to others looking into IBM HashiCorp Infrastructure Lifecycle Management that they define their infrastructure standards, module structure, and state management approach before rolling it out widely. Also, start with a few repeatable use cases, integrate it with your CI/CD pipeline, and gradually expand once the team is comfortable with Terraform.

“Overall, IBM HashiCorp Infrastructure Lifecycle Management has been a practical solution for standardizing infrastructure management and reducing manual provisioning work. The main thing I would emphasize is to invest time upfront in module design, state management, and governance, because that makes the platform much easier to scale later. I would rate this solution eight out of ten overall..”

Verified user

Developer at a healthcare company with 1,001-5,000 employees

[Read full review](#) 

“I would say IBM HashiCorp Infrastructure Lifecycle Management deserves a nine out of ten marks. With the improvement I mentioned for platform teams, that would be really valuable. Apart from that, overall, it definitely deserves this rating.

“I have not worked with IBM HashiCorp Infrastructure Lifecycle Management's AI capabilities and have not explored them yet. Most of the time, we used [GitHub](#) Copilot or Codex to generate code for Terraform. My experience with the AI capabilities of Terraform is limited.

“For anyone using HashiCorp Terraform, I would suggest using reusable templates and Terraform modules to minimize code repeatability..”

Gardihewagesathira Gardihewa

DevOps Engineer at a consumer goods company with 10,001+ employees

[Read full review](#) 

About this buyer's guide

Thanks for downloading this PeerSpot report.

The summaries, overviews and recaps in this report are all based on real user feedback and reviews collected by PeerSpot's team. Every reviewer on PeerSpot has been authenticated with our triple authentication process. This is done to ensure that every review provided is an unbiased review from a real user.

Get a custom version of this report... Personalized for you!

Please note that this is a generic report based on reviews and opinions from the collective PeerSpot community. We offer a [customized report](#) of solutions recommended for you based on:

- Your industry
- Company size
- Which solutions you're already considering

The customized report will include recommendations for you based on what other people like you are using and researching.

Answer a few questions in our short wizard to get your customized report.

[Get your personalized report here](#)

About PeerSpot

PeerSpot is the leading review site for cloud, AI, and business software. We created PeerSpot to provide a trusted platform to share information about software, applications, and services. Since 2012, over 22 million people have used PeerSpot to choose the right software for their business.

PeerSpot helps tech professionals by providing:

- A list of products recommended by real users
- In-depth reviews, including pros and cons
- Specific information to help you choose the best vendor for your needs

Use PeerSpot to:

- Read and post reviews of products
- Access over 30,000 buyer's guides and comparison reports
- Request or share information about functionality, quality, and pricing

Join PeerSpot to connect with peers to help you:

- Get immediate answers to questions
- Validate vendor claims
- Exchange tips for getting the best deals with vendor

Visit PeerSpot: www.peerspot.com

PeerSpot

244 5th Avenue, Suite R-230 • New York, NY 10001

reports@peerspot.com

+1 646.328.1944