INFRASTRUCTURE AUTOMATION IN OPENSTACK

Marek MORAVCIK – Lubica ZIDEKOVA

University of Žilina, Faculty of Management Science and Informatics, Slovak Republic

e-mail: marek.moravcik@fri.uniza.sk, lubica.zidekova@fri.uniza.sk

Abstract

Cloud infrastructure orchestration is a key element of a modern application deployment and management environment. As an open platform for building private and public clouds, OpenStack enables multiple automation paths through declarative tools. This article compares the three most widely used approaches to OpenStack orchestration: the native Heat module, the general-purpose Terraform tool, and its open-source derivative OpenTofu. The comparison focuses on architecture, infrastructure definition language, extensibility, compatibility, and performance. The results indicate that while Heat provides the deepest integration with the OpenStack API, Terraform and OpenTofu offer greater portability and ecosystem flexibility.

Keywords**:** *OpenStack, automation, Heat, Terraform, OpenTofu*

1 INTRODUCTION

OpenStack is a comprehensive set of services that enable the management of compute, network, and storage resources. With the increasing level of automation, the need for effective tools for Infrastructure as Code (IaC) is becoming a necessity.

The concept of Infrastructure as Code (IaC) enables automated, reproducible, and versionable infrastructure deployment through declarative configurations. The most widely used open-source tools in this area include OpenStack Heat, Terraform, and the newly created OpenTofu project (a fork of Terraform). Although all three solutions pursue the same goal - to enable infrastructure management using code - they differ in architecture, scope, and design philosophy.

The goal of this paper is to provide a systematic comparison of three approaches to orchestrating OpenStack infrastructures: Heat, Terraform, and OpenTofu (a fork of Terraform after HashiCorp changed its license to BUSL-1.1) [1] [2].

2 OPENSTACK AUTOMATION

OpenStack [3] is an open-source cloud platform that allows you to build and manage private and public clouds similar to those provided by large commercial providers (such as AWS, Azure, or Google Cloud). It is a modular system consisting of several interconnected components (services) that together cover all the basic areas of cloud infrastructure - computing power, storage, networking, identity, monitoring, and orchestration.

There are not many scientific papers dealing with OpenStack environment automation. In [4] and [5], authors ate comparing mainly application automation with tools like Puppet, Ansible, or Chef. Authors in the [6] are automating OpenStack deployment in multicloud environments. In our paper, we will deal with automation of environment for VMs in installed and running OpenStack.

Currently, we have several ways of OpenStack automation. We can create shell scripts that use basic CLI commands. This option is not very scalable, and the biggest drawback is that the user must have access to the OpenStack command line, which is usually undesirable.

The second option is to use the API that OpenStack offers. This is a slightly better option, but the user needs to follow the API documentation, which can and often does change between different versions of OpenStack.

The third option is to use automation scripts, or automation languages, such as Heat, Terraform, and OpenTofu, which we will look at in more detail in the following sections.

2.1 OpenStack Heat

OpenStack Heat [7] is an orchestrator developed as native part of the OpenStack ecosystem. It allows the description of infrastructure through templates in YAML [8] format (Heat Orchestration Template – HOT), which define resources such as virtual machines, networks, storage volumes and dependencies between them. Heat is tightly integrated with the OpenStack API and is therefore primarily intended for managing cloud infrastructure in an OpenStack environment. Its advantage is native integration with other OpenStack components (Nova, Neutron, Cinder, Glance), but at the same time it is strongly tied to this environment and its usability outside of OpenStack is limited. Heat is oriented towards orchestration within a single cloud and does not provide multi-cloud functionality.

2.2 Terraform

Terraform [1] from HashiCorp is the de facto standard in the field of IaC tools. It uses HCL's own declarative language (HashiCorp Configuration Language) and supports a wide range of providers - from public clouds (AWS, Azure, GCP) to private infrastructures and various third-party services. Terraform allows for change planning (the so-called plan phase), which provides an overview of proposed modifications before they are applied. Thanks to its modular system, it is highly extensible and portable between environments. Compared to Heat, it provides much broader multi-cloud support

and flexibility but is not natively connected to the OpenStack API as deeply as Heat. Terraform was originally open-source, but since 2023 it has switched to the Business Source License (BSL), which has provoked community reactions and the emergence of its open-source fork - OpenTofu.

2.3 OpenTofu

OpenTofu [2] is a fork of Terraform maintained by the Linux Foundation in response to the Terraform license change. It maintains compatibility with existing HCL configurations and providers, while the project aims to maintain a fully open development model and community governance. Functionally, OpenTofu is currently almost identical to Terraform, but gradual divergence in security, extensibility, and integration with cloud-native tools is expected. OpenTofu is positioned as a long-term open-source alternative to Terraform without the restrictions of a commercial license.

3 ORCHESTRATION SCRIPTS COMPARISON

We choose 5 criteria to compare: Integration with OpenStack, support for multiple cloud platforms, modularity and add-on capabilities of various components, documentation accessibility and the last criterion is a license for home and commercial use.

OpenStack integration

As the name suggests, OpenStack Heat is a direct part of the OpenStack ecosystem. It uses internal API calls, making the system of modules and API calls transparent to users. It has direct integration into the Horizon dashboard, so the user does not need to allow any additional communication. When the OpenStack system is updated, the Heat module is also updated in the event of a change in the API calls, which guarantees a smooth and transparent transition to the new version for the user.

Terraform and OpenTofu use the API calls of individual OpenStack modules. From a network perspective, it is necessary to publish an API interface for each OpenStack module to the user, which creates a possible risk of API interface abuse. If the OpenStack system is composed of multiple modules, this can pose a major security risk. Not to mention that when updating a module and possibly changing the API interface, it is necessary to modify and update the underlying Terraform or OpenTofu runtime.

Multicloud support

OpenStack Heat is designed solely for use with OpenStack. It cannot be used with any other cloud provider or platform.

On the other hand, Terraform and OpenTofu are designed for usage with multiple cloud platforms. Although each cloud provider uses its own logical structure and it is necessary to modify the script for a specific provider, the overall logic for any provider is the same.

Modularity

One of the advantages over OpenStack Heat is the ability to define variables and native support for using multiple text files, which makes structuring and navigating files much easier when defining larger environments.

Documentation

All three tools are well documented, and documentation is publicly available. OpenStack [9], Terraform [10] and also OpenTofu [11] documentations has explanations, sample usages, are well structured and also have separate documentations per released versions.

License / openness

OpenStack Heat is, like all OpenStack modules, open source and released under Apache License 2.0. It is developed by community, and everyone has access to source code and can use it freely, by the terms of license.

Terraform's license changed from the Mozilla Public License (MPL) 2.0 to the Business Source License (BSL) 1.1 in August 2023 for new versions [1]. The BSL allows for free non-production use and commercial use, but it prohibits offering a competitive service built on future Terraform releases.

OpenTofu is licensed under the permissive Mozilla Public License 2.0 (MPL 2.0) [2], making it fully open source. The OpenTofu is community-governed and has no commercial restrictions.

Final comparison

Table 1 contains the actual comparison of the tools based on the criteria we defined. We rated each criterion with points, with 1 being the worst and 5 being the best result.

Table 1

Criteria	Heat	Terraform	OpenTofu
OpenStack integration	5	4	4
Multicloud support	1	5	5
Modularity	2	5	5
Documentation	5	5	5
License / openness	5	2	5

From the table above, we can say that there is no clear winner. Each tool has its pros and cons and may be better than its opponents in certain types of use.

4 CONCLUSION

The comparison shows that the choice of orchestrator depends on the deployment context. Heat is ideal for pure OpenStack environments that require native integration and direct binding to internal services.

Terraform provides a high level of abstraction and is suitable for hybrid and multicloud scenarios, but its licensing model limits open use.

OpenTofu is a promising alternative with the same functionality as Terraform, while maintaining open development and long-term independence.

From a practical point of view, Heat is recommended for projects closely tied to OpenStack, while OpenTofu is a more suitable choice for organizations that prefer open and multiplatform environments.

Acknowledgement: Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00010.

REFERENCES

[1] HashiCorp, „Terraform,“ [Online]. Available: https://developer.hashicorp.com/terraform. [Cit. 1 11 2025].

[2] Linux Foundation, „OpenTofu,“ [Online]. Available: https://opentofu.org/. [Cit. 1 11 2025].

[3] OpenStack Foundation, „OpenStack,“ [Online]. Available: https://www.openstack.org/. [Cit. 1 11 2025].

[4] P. MANDAL and R. JAIN, „Comparison of OpenStack Orchestration Procedures,“ rev. *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, Sharm El Sheikh, Egypt, 2019.

[5] D. VLADUSIC and D. RADOLOVIC, „Infrastructure as Code for Heterogeneous Computing,“ rev. *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, Timisoara, Romania, 2020.

[6] G. TRICOMI, A. PANARELLO, G. MERLINO, F. LONGO, D. BRUNEO and A. PULIAFITO, „Orchestrated Multi-Cloud Application Deployment in OpenStack with TOSCA,“ rev. *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, Hong Kong, China, 2017.

[7] Openstack Foundation, „OpenStack Heat documentation,“ [Online]. Available: https://wiki.openstack.org/wiki/Heat. [Cit. 1 11 2025].

[8] O. BEN-KIKI, C. EVANS A I. D. NET, „YAML Ain't Markup Language version 1.2.2,“ 1 10 2021. [Online]. Available: https://yaml.org/spec/1.2.2/. [Cit. 1 11 2025].

[9] OpenStack Foundation, „OpenStack Resource Types,“ [Online]. Available: https://docs.openstack.org/heat/latest/template_guide/openstack.html. [Cit. 2 11 2025].

[10] Hashicorp, „Terraform OpenStack provider,“ [Online]. Available: https://registry.terraform.io/providers/terraform-provider-openstack/openstack/latest/docs. [Cit. 2 11 2025].

[11] Linux Foundation, „OpenTofu OpenStack provider,“ [Online]. Available: https://search.opentofu.org/provider/terraform-provider-openstack/openstack/v2.1.0. [Cit. 2 11 2025].