

SOLUTION BRIEF

Executive Summary

IT solution providers need to satisfy the demand for the delivery of diverse services in the short-term. Time-to-market when offering IT solutions, IaaS or just provisioning infrastructure is a key factor for the company's competitiveness. This requires companies to utilize tools capable of providing the agility to deliver new features, scalability, and infrastructure anywhere almost instantly.

Historically, managing IT infrastructure has been a manual process, which led to increased operational costs, lack of instant scalability and inconsistencies. With the advent of the concept of Infrastructure as code, all these challenges can now be easily overcome, with leading to benefits like simplicity, accountability and improved efficiency.

Terraform is an open-source infrastructure-as-code tool created by Hashicorp and licensed through the Apache 2 License. By using Terraform, end users can easily create complex infrastructures anywhere, since Terraform has a huge list of providers supported, including Apache CloudStack.

The latest release of the Apache CloudStack Terraform Provider, 0.4.0, is the first release under the Apache 2 license.

This release includes the following changes:

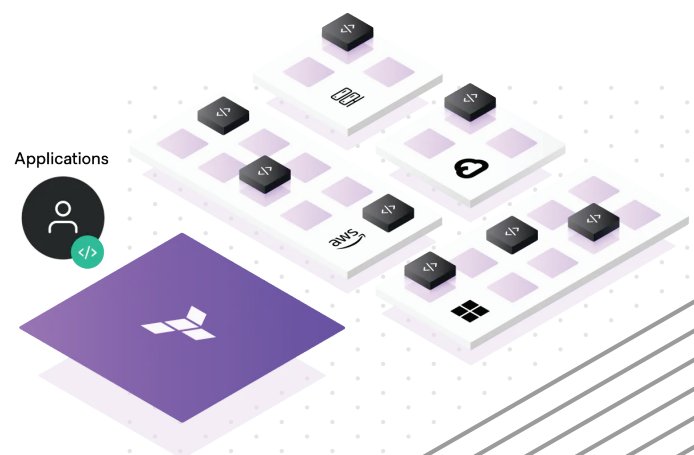
- Relicensing with Apache License v2.0
- Validated and fixed all existing resource creations to work as expected
- Updated documentation on using and also developing the provider
- Bug fixes on syncing the data from CloudStack
- Fixed resource destroy using Terraform for some resources

Terraform converts the Apache CloudStack API into declarative configuration files and provides the agility needed to serve different use cases on top of Apache CloudStack.

Terraform - Automate Infrastructure on Any Cloud

Terraform is an **open-source infrastructure as code software** tool that provides a consistent CLI workflow to **manage hundreds of cloud services**. Terraform codifies cloud APIs into declarative configuration files.

Terraform allows infrastructure to be expressed as code in a simple, human-readable language called HCL (HashiCorp Configuration Language). It reads configuration files and provides an execution plan of changes, which can be reviewed for safety and then applied and provisioned. **Extensible providers allow Terraform to manage a broad range of resources, including IaaS, PaaS, SaaS, and hardware services.**



SOLUTION BRIEF

CloudStack – A Turnkey Solution for Cloud and IaaS Builders



Apache CloudStack is the **leading open-source cloud orchestration platform**. It is in use by many of the world’s largest public and private clouds. CloudStack is a multi-hypervisor, multi-tenant, high-availability **Infrastructure as a Service cloud management platform**.

Apache CloudStack is software that provides a cloud orchestration layer, giving **automation of the creation, provisioning and configuration of IaaS components** (such as virtual servers). It turns an existing virtual infrastructure into a cloud-based infrastructure as a Service (IaaS) platform. The fact CloudStack leverages existing infrastructure means that the **cost and time for an organisation to build a multi-tenant IaaS platform is greatly reduced**.

Architecture – CloudStack and Terraform Integration



Terraform relies on a set of plugins known as Providers in order to provision or manage resources and cloud services. As part of the integration of Terraform and Apache CloudStack, Terraform requires a specific CloudStack provider, which acts as a transition layer between Terraform and Apache CloudStack. This provider was written to provision and manage resources such as virtual machines, networks, templates, volumes etc., using the CloudStack APIs.

There are two main working components here:

- **Terraform Core**
- **CloudStack’s Provider**

Terraform core is an engine which requires a configuration file (*.tf) as an input. In this file users define which cloud resources need to be provisioned and managed.

This is where the HashiCorp Configuration Language is used to prepare the configuration file.

Terraform also uses the state file (*.tfstate) as input. This file holds the state information of the resources created or managed by Terraform. This is created by Terraform core itself based on the initial configuration. Afterwards, Terraform syncs with CloudStack and keeps this state file up to date. This data is used to create action plans regarding what needs to be done in CloudStack whenever the user tries to apply the configuration.

Once the user creates and confirms the action plan, Terraform uses the provider to apply these actions in Apache CloudStack. As a result, once those are completed, Terraform fetches the resource states and saves it to its state file.

SOLUTION BRIEF



“We are excited to see this integration between Apache CloudStack and Terraform. We observe an increasing number of use cases for CloudStack that are being driven by infrastructure automation and infrastructure repeatability.

Organizations also seek consistency across any cloud platform: whether that is their internal infrastructure driven by CloudStack or major public clouds. Terraform, along with CloudStack, is a perfect fit for these use-cases!”

Giles Sirett
CEO of ShapeBlue
Chairman of CloudStack European User Group



Technical Benefits

- Manage infrastructure across clouds
- Reproduce infrastructure easily across hybrid environments
- Adding an automation layer on top of Apache CloudStack, making cloud deployment and maintenance much easier
- Increased visibility on infrastructure changes



Business Benefits

- Open-source solution
- Decreased time in deploying infrastructure
- Decreased operational costs
- Simple management of large infrastructures
- Strong technical community



CloudStack and Terraform bring scalability and flexibility. The immediate benefit out of them is that you can have tested and proven blueprints and roll out environments quickly. Terraform is perfect for quickly creating Test/Dev environments.

In distributed teams, collaboration is extremely important. Infrastructure as code is a huge boost, helping teams to collaborate on code.

Tamara Muryshkin
Enterprise Service Manager at Fraunhofer-Gesellschaft

SOLUTION BRIEF

Use Cases

From the end-user perspective, service providers can offer a way for users to design and deploy their own cloud infrastructure as code, more easily, using the Terraform CloudStack provider.

CI/CD

From a business continuity point of view, once the virtual infrastructure is designed and deployed, if any changes are required, users need to add the necessary code with the desired changes to the configuration file and apply it by using the Terraform CloudStack provider. Terraform will take care of the changes without interrupting the services by adding the changes contained in the code. This enables users to have a CI/CD policy maintaining control over assets in the cloud.

Integration of Multiple Resources from a Single Point

As Terraform has a rich ecosystem of providers, users can manage public DNS entries of load balancer's IP addresses created on CloudStack. They can also manage external firewalls when using shared networks, add CloudStack instances in their monitoring system and many more.

All these resources will be kept in the Terraform configuration file and can be easily managed from a single point.



 **Fraunhofer**



Terraform is perfect for all innovative early adopters. It is the beginning of large journey, which is recommended to explore.

Future versions of this Terraform plugin should allow an end-to-end infrastructure as code lifecycle for Apache CloudStack infrastructure.

Tamara Muryshkin
Enterprise Service Manager at Fraunhofer-Gesellschaft

Key Outputs

- Best fit for orchestrating cloud services like CloudStack, setting up cloud infrastructure from scratch and managing the existing cloud.

- Ensures that an environment is in its desired state continuously.

Resources

<https://github.com/apache/cloudstack-terraform-provider/wiki>
<https://registry.terraform.io/providers/cloudstack/cloudstack/latest/docs>
<https://cloudstack.apache.org/>

*"Apache", "CloudStack", "Apache CloudStack", the Apache CloudStack logo, the Apache CloudStack Cloud Monkey logo and the Apache feather logos are registered trademarks or trademarks of The Apache Software Foundation.

