

Managing Kubernetes Applications Through Terraform and AWS EKS



The advent of automation and cloud technology has become more efficient, simpler, and more reliable in providing complex infrastructure. For Instance, APIs can automate the resources in [cloud platforms](#) such as [Microsoft Azure](#), [Google Cloud Platform \(GCP\)](#), and [Amazon Web Services \(AWS\)](#). Moreover, your application's underlying architecture and configuration may now automate by using code.

[Infrastructure as Code \(IaC\)](#) permits users to define, deploy, update, and delete infrastructure by writing and executing the code. This approach treats all operations as software even though that represents the hardware configuration. As a result, organizations can reduce the risk of configuration drift and achieve more dependable outcomes in their architectures if they run IaC tools to develop their application environments.

Terraform is the first service tool that springs to mind regarding infrastructure provisioning, including the AWS EKS Cluster. Terraform is considerably easier to learn

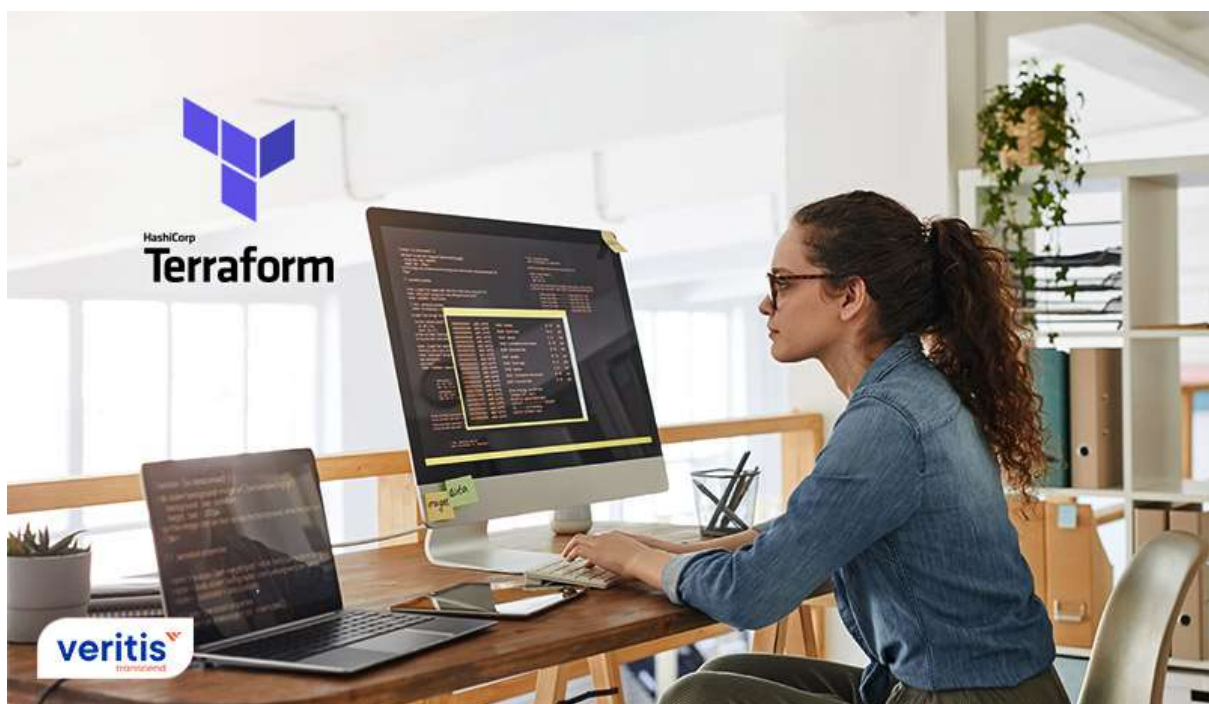
than manually putting up infrastructure. Terraform is one of the most popular IaC tools and an open-source, cloud-agnostic provisioning tool platform used to change, build and version infrastructure securely and efficiently. It's convenient for setting up sophisticated platforms like Kubernetes clusters, which have become more popular as cloud-native solutions gain traction.

To reduce the complexity of cluster management, [cloud service providers](#) such as AWS have established managed services like Amazon EKS (Amazon Elastic Container Service for Kubernetes). AWS is wholly responsible for running, managing, provisioning, and auto-scaling the nodes. It is the master across a wide range of AWS Availability Zones (AZs) and ensures high availability. Unless they adopt the Fargate serverless engine, users are entirely responsible for managing and adding the EC2 worker nodes.

Before going in-depth info about how to deploy a Kubernetes application with AWS EKS and terraform, let's explore the concepts first – what is Terraform, the advantages of Terraform, how does terraform work, and what is AWS EKS.

Useful Link: [How to Optimize Kubernetes Autoscaler to Better Business](#)

What is Terraform?





Terraform is an open-source 'Infrastructure as code' software application that allows users to create, alter and improve infrastructure safely and reasonably. Users can use a consistent workflow to provide and manage the infrastructure throughout its lifecycle. Terraform can manage both low-level and high-level components, such as compute, storage, and networking resources for low-level and DNS records as well as SaaS services for high levels.

Terraform for infrastructure as a code software platform employs declarative configure files to codify cloud APIs and permit users to manage hundreds of cloud services using a standard CLI approach. The Amazon Elastic Kubernetes Service (EKS) is a managed Kubernetes service responsible for scaling, deploying, and managing the containerized applications with Kubernetes in the cloud or on-premises.

Terraform is an IaC tool that defines both on-premises and cloud resources in human-readable configuration files that users can reuse and share. In addition, Infrastructure as Code (IaC) approaches use multiple ways to manage modern infrastructures with tools such as AWS Cloud Development Kit (CDK), Hashicorp's Terraform, and AWS CloudFormation.

How does Terraform work?

Terraform operates application programming interfaces to create and manage resources on the cloud platform and other services (APIs). Thanks to providers who allow Terraform to work, this application can work with almost any platform or service that has an accessible API.

HashiCorp and the Terraform community have built more than 1700 providers to handle thousands of various types of resources and services, and the number continues to increase. The Terraform Registry lists all available service providers, such as Microsoft Azure, Amazon Web Service (AWS), Google Cloud Platform (GCP), DataDog, GitHub, [Kubernetes](#), Helm, Splunk, and many more to the list.

Headquarters: Veritis Group, Inc , 1231 Greenway Drive, Suite 1040, Irving, TX 75038

Phone: 972-753-0022 | **Email:** connect@veritis.com

There are three stages for the Terraform workflow:

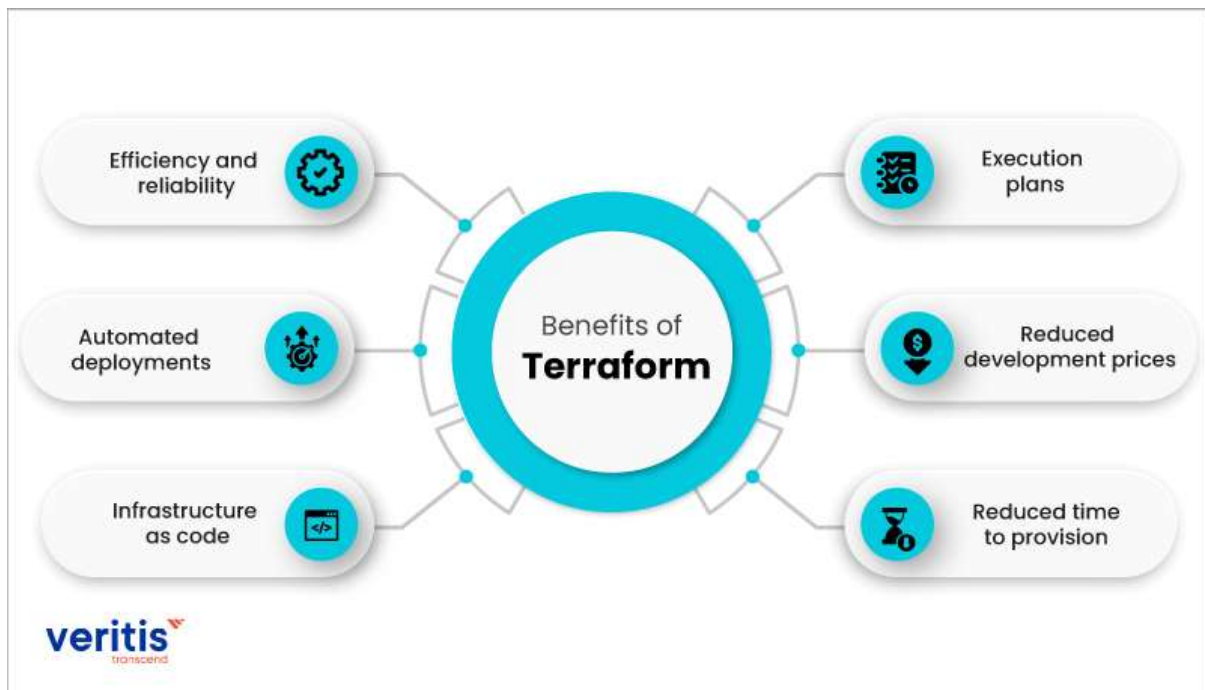
1) Write: Users create resources that come across numerous cloud services and providers. For instance, in a Virtual Private Cloud (VPC) network with security groups and a load balancer, you might develop a configuration to deploy an application on virtual machines.

2) Plan: Terraform generates an execution plan that defines the infrastructure, and it will create, update, or delete depending on your configuration's existing infrastructure.

3) Apply: Terraform executes the proposed activities in the correct order, considering any resource dependencies. For instance, if you modify the number of virtual machines in a VPC and update its characteristics. Terraform will rediscover the VPC before reducing the virtual machines.

Useful Link: [Top 10 DevOps Tools to Pick for Your Business](#)

Benefits of Terraform



Terraform can manage custom in-house solutions as well as popular and existing service providers. Operating Terraform to provision the infrastructure has numerous advantages for EKS clusters, including other resources managed through IaC.

A) Efficiency and reliability

An automated deployment technique allows significantly faster infrastructure changes. Terraform executes the intended state by running underlying API requests in the correct order, with the flexibility to specify resource prerequisite requirements.

B) Automated deployments

You can run CI/CD approaches to automating the provisioning of your infrastructure if your infrastructure is defined as code. Manual deployments by individual personnel are no longer required with [CI/CD pipelines](#). Instead, more people can be empowered to activate pipelines without requiring a technical understanding of manual processes. Moreover, your infrastructure as code can have rich quality checks like automated testing in the CI stage of the pipeline as well as pull requests for changes to the source code.

C) Infrastructure as code

Infrastructure as code, rather as physical hardware setup or interactive configuration tools, controls, and provision computer data centres through machine-readable specification files. Terraform permits users to run operating coding techniques to construct, change, and version their infrastructure. It will allow you to design your infrastructure or apps using configuration files, as well as store the status of the user's infrastructure for later updates or modifications.

D) Execution plans

The Terraform plan command executes a set of execution instructions. By default, creating a project involves reading any existing remote objects' present state to ensure that the Terraform state is currently up to date. In addition, it notes the differences between the current configuration and the initial state.

Terraform features a stage called planning that develops an execution plan. Then, when you apply for a call, the execution plan displays how Terraform will perform. This allows you to avoid any Terraform surprises.

E) Reduced development prices

Creating deployment environments and on-demand development can help you save money. Moreover, you can evaluate such situations prior to updating system changes.

F) Reduced time to provision

A company's traditional click-ops deployment processes can take days or even weeks, prone to errors. Full deployment with Terraform can take as little as a few minutes. For instance, you can standardize provision of numerous Alibaba Cloud services at once. Migrations, as well as new deployments can be completed efficiently and rapidly.

Useful Link: [Top 15 AWS Machine Learning Tools in the Cloud](#)

Terraform with Kubernetes





The concepts of Terraform and Kubernetes are pretty perplexed. But actually, they both work together for output, and they are not alternatives.

Kubernetes, also known as K8s, is an extensible, portable, open-source container orchestration platform built to automate containerized applications' scaling, management, and deployment. Indeed, Kubernetes has become the de-facto standard for container orchestration. It is the flagship project of the Cloud Native Computing Foundation (CNCF), backed by top MNCs such as Microsoft, IBM, Google, AWS, Intel, Red Hat, and Cisco.

On the other hand, Terraform is a massive infrastructure as a code solution that permits developers to automate the whole infrastructure across many public and private clouds. This application automates and manages the Software as a Service (SaaS), Platform as a Service (PaaS), or even Infrastructure as a Service (IaaS) level abilities. All of these resources are in parallel across all of those service providers. Terraform can automate the provisioning of Kubernetes clusters, especially the deployment of applications into a cluster and managed Kubernetes clusters on the cloud platforms.

Deploying Kubernetes Applications with Terraform

Design the pattern for the Kubernetes provider by creating a namespace, then deploying the application in a pod, and revealing the pod to users as a service at the end, which is the ideal way to work with or interact with Terraform Kubernetes provider.

Meanwhile, you'll need to have your Kubernetes cluster up and operating to do all of this. The four steps are outlined below:

- 1) Setup our Kubernetes service provider
- 2) Pod Deployment
- 3) Display the Pod with a service
- 4) Finally, verify the app is running

Headquarters: Veritis Group, Inc , 1231 Greenway Drive, Suite 1040, Irving, TX 75038

Phone: 972-753-0022 | **Email:** connect@veritis.com

Useful Link: [How to Optimize Kubernetes Autoscaler to Better Business](#)

What is Amazon EKS?



Amazon Elastic Kubernetes Service (AWS EKS) is a service platform that manages and provisions the control panel of your Kubernetes applications in the on-premises or cloud. This means you don't need to worry about the upgrades, high availability, and the security of your control panel.

It operates your Kubernetes management infrastructure across numerous AWS Availability Zones to get rid of a single point of failure. AWS EKS is Kubernetes-compliant; users may allow running existing Kubernetes plugins and tools from the community and partner. The application that runs on any regular Kubernetes environment is entirely compatible with AWS EKS and can be migrated.

AWS ECR

Amazon Elastic Container Registry (ECR) is an AWS product that manages, stores, and deploys Docker container registry service. Users' container images are stored in a highly available and scalable architecture on an Amazon ECR private registry. ECR supports

public and private repositories, which are more secure. By running AWS CLI, users can either pull or push pics to ECR.

Deploying a Kubernetes Applications using AWS EKS



- By using Node.js, you have to create a simple web application
- Develop a web application Docker image
- When we push the Docker image, AWS ECR will receive it
- For the EKS cluster, develop a VPC by using public and private subnets
- Develop a Kubernetes cluster
- Develop Kubernetes workers like private and public workers
- Using Kubernetes, deploy your web application

Prerequisites

Users should have an active AWS account with a subscription and allow to access your AWS IAM account with credentials. Create an AWS account and an IAM user in your AWS account if you don't already have one.

The most recent version of the AWS command-line interface (CLI) should be installed at a location in your system path.

Useful Link: [Differences Between Deep Learning and Machine Learning](#)

Conclusion

Elastic Kubernetes Service (EKS), a managed Kubernetes cluster, reduce the complexity and overhead of provision and optimize the control planes for software teams. Moreover, automation technology such as IaC improves the infrastructure management life cycle with numerous advantages. However, if there are no tools to allow pertinent insight, reporting, and monitoring for optimal usage, Kubernetes cluster management in AWS might have substantial economic implications.

Terraform is a significant IaC tool that permits users to create, version control, and modify the cloud and on-premises resources efficiently and securely. Terraform Kubernetes deployment can eliminate human errors and save time when running a traditional model for application deployment. Users can run the Terraform Kubernetes Deployment method to develop an EKS cluster with Terraform while automating the EKS cluster with the creation process and having additional control over the complete infrastructure management process via code.

Deploying a Kubernetes application with AWS EKS and Terraform is a complicated task. This is why companies seek Veritis's services. Veritis, the [Stevie Award Winner](#), guides software teams and organizations by offering multiple services from cloud to technology advisory with cost-effective solutions.

Services