

Terraform: Infrastructure as Code

Terraform Use-cases

Terraform Use-cases

Heroku App Setup

Terraform can be used to codify the setup required for a Heroku application, ensuring that all the required add-ons are available, configuring DNSimple to set a CNAME, or setting up Cloudflare as a CDN for the app without using a web interface.

src: <https://www.terraform.io/intro/use-cases.html>

Terraform Use-cases

Multi-Tier Applications

With Terraform each tier of N-tier architecture can be described as a collection of resources, and the dependencies between each tier are handled automatically; Terraform will ensure the database tier is available before the web servers are started.

src: <https://www.terraform.io/intro/use-cases.html>

Terraform Use-cases

Self-Service Clusters

Terraform configurations can be shared within an organization enabling customer teams to use the configuration as a black box and use Terraform as a tool to manage their services.

src: <https://www.terraform.io/intro/use-cases.html>

Terraform Use-cases

Software Demos

Software writers can provide a Terraform configuration to create, provision and bootstrap a demo on cloud providers like AWS, which allows users to easily demo the software on their own infrastructure.

src: <https://www.terraform.io/intro/use-cases.html>

Terraform Use-cases

Disposable Environments

Using Terraform, the production environment can be codified and then shared with staging, QA or dev. These configurations can be used to rapidly spin up new environments to test in, and then be easily disposed of, which can help to maintain parallel environments.

src: <https://www.terraform.io/intro/use-cases.html>

Terraform Use-cases

Software Defined Networking

Terraform can be used to codify the configuration for software defined networks, which can then be used by Terraform to automatically setup and modify settings by interfacing with the control layer.

src: <https://www.terraform.io/intro/use-cases.html>

Terraform Use-cases

Resource Schedulers

Resource schedulers can be treated as a provider, this allows Terraform to be used in layers: to setup the physical infrastructure running the schedulers as well as provisioning onto the scheduled grid.

src: <https://www.terraform.io/intro/use-cases.html>