Terraform Docker

Sunday, October 29, 2023

5:12 AM

# Last updated 10/29/2023

Using GitHub to sign into terraform

[Docker | Terraform | HashiCorp Developer](https://developer.hashicorp.com/terraform/tutorials/docker-get-started)

* + Terraform can manage infrastructure on multiple cloud platforms.
  + The human-readable configuration language helps you write infrastructure code quickly.
  + Terraform's state allows you to track resource changes throughout your deployments.
  + You can commit your configurations to version control to safely collaborate on infrastruct

Terraform plugins called providers let Terraform interact with cloud platforms and other services via their application programming interfaces (APIs).

# [Standardize your deployment workflow](https://developer.hashicorp.com/terraform/tutorials/docker-get-started/infrastructure-as-code#standardize-your-deployment-workflow)

To deploy infrastructure with Terraform:

* + Scope - Identify the infrastructure for your project.
  + Author - Write the configuration for your infrastructure.
  + Initialize - Install the plugins Terraform needs to manage the infrastructure.
  + Plan - Preview the changes Terraform will make to match your configuration.
  + Apply - Make the planned changes.

# [Track your infrastructure](https://developer.hashicorp.com/terraform/tutorials/docker-get-started/infrastructure-as-code#track-your-infrastructure)

Terraform keeps track of your real infrastructure in a state file, which acts as a source of truth for your environment. Terraform uses the state file to determine the changes to make to your infrastructure so that it will match your configuration.

# [Collaborate](https://developer.hashicorp.com/terraform/tutorials/docker-get-started/infrastructure-as-code#collaborate)

Terraform allows you to collaborate on your infrastructure with its remote state backends. When you use Terraform Cloud (free for up to five users), you can securely share your state with your teammates, provide a stable environment for Terraform to run in, and prevent race conditions when multiple people make configuration changes at once.

You can also connect Terraform Cloud to version control systems (VCSs) like GitHub, GitLab, and others, allowing it to automatically propose infrastructure changes when you commit configuration changes to VCS. This lets you manage changes to your infrastructure through version control, as you would with application code.

# Install terraform

brew tap hashicorp/tap

brew install hashicorp/tap/terraform

terraform -version

Terraform v1.6.2

on linux\_amd64

terraform -help

terraform -help plan

terraform -install-autocomplete

# Create NGINX Container

cd $GIT\_ROOT

mkdir terraform

cd terraform/

mkdir learn-terraform-docker-container

cd learn-terraform-docker-container

nvim main.tf

terraform init

terraform plan

terraform apply

Run browser <http://localhost:8000/>

docker ps

terraform destroy