TERRAFORM – VS – CLOUDFORMATION

▸ Terraform was inspired by CloudFormation

  ▸ Infrastructure codified into a configuration file, allowing infrastructure to be elastically created, updated, destroyed

▸ Terraform goes a bit further by being cloud-agnostic and allowing multiple providers and services to be composed together. In a unified syntax.

▸ Terraform separates the planning phase from the execution phase.

TERRAFORM – VS – BOTO

▸ Boto and the like, are libraries used to provide native access to different cloud providers and services through their APIs. This gives the developer low-level programmatic access, but often means they will need to create their own tooling to build and manage their infrastructure.

▸ Terraform isn't intended to give such low-level access, but instead a higher level syntax for describing the resources and services that should be created.

▸ With the flexible plugin-based models, Terraform can support almost any service that exposes API's