Terraform

Terraform

Infrastructure as Code

- Provisioning infrastructure through software to achieve consistent and predictable environments.

Core Concepts

- Defined in code
- Stored in source control
- Declarative vs Imperative
- Idempotent and Consistent
- Push or Pull

Imperative

- Imperative is about the HOW
- For example, if I was writing an imperative program for building a house, it would go something like this:
  - Build the foundation
  - Put in the framework
  - Install the utilities
  - Add the walls
  - Finishing touches

Declarative

- Declarative is about the WHAT
- Building a house declaratively would include the following steps:
  - I don’t care how you build it, but I want a nice fireplace, a lakefront view, and a big kitchen.

In this declarative program, I have told you the outputs that I want. I know that if I give you inputs in the form of money, I will get the desired outputs.

Infrastructure as Code benefits

- Automated deployment
- Consistent environments
- Repeatable process
- Reusable components
- Documented architecture

Install Terraform

- Windows:
  - https://www.decodingdevops.com/how-to-install-terraform-on-windows-10-or-8-or-7/
- Linux:
  - https://www.decodingdevops.com/how-to-install-terraform-on-linux-servers/
- Mac:
  - https://www.bonusbits.com/wiki/HowTo:Install_Terraform_on_macOS