**Terraform – Getting Started**

**Infrastructure as Code:** Provisioning infrastructure through software to achieve consistent and predictable environments. Environment will look how the configuration file says it should.

Think Azure DevOps yaml pipeline files.

**Declarative vs Imperative:** Imperative commands describe what you need and how to get there. Declarative language defines what you want, its components, and itll build it for you. Terraform is a declaration based software.

**Idempotent:** Desired state vs actual state idea: If a command is executed and the process is already in effect, it will decline the command. Terraform pushes its commands to target machines.

**Automated deployments, consistent environments, repeatable process, reusable components, documented architecture.**

Scenario:

Say you want to deploy frontend and backend applications to Azure and you need to provision resources for it…