Terraform Architecture & Concept Map

Terraform is an Infrastructure as Code (IaC) tool that allows you to define, provision, and manage cloud infrastructure through code.

This PDF provides a conceptual overview and architectural visualization of Terraform components and their workflow, especially focused on AWS integration.

Core Terraform Components

- Provider: Plugins that allow Terraform to interact with cloud services like AWS, Azure, or GCP.
- Resources: The actual infrastructure items (EC2, S3, VPC) defined in Terraform configuration files.
- Variables: Used to make code dynamic and reusable by assigning flexible input values.
- Outputs: Display important information after Terraform applies changes (e.g., EC2 IP).
- State File: Terraform's memory of your infrastructure stored in 'terraform.tfstate'.
- Modules: Reusable Terraform code blocks that help structure large deployments.
- Workspaces: Used to manage multiple environments like Dev, Test, and Prod.
- Provisioners: Used to run configuration scripts on created instances (e.g., install NGINX).
- Backends: Where Terraform state is stored ? can be local or remote (S3, Terraform Cloud).

Terraform Architecture Flow

1. Developer writes Terraform code (main.tf, variables.tf, outputs.tf).
2. Terraform connects to cloud via the Provider (e.g., AWS).
3. 'terraform init' downloads provider plugins.
4. 'terraform plan' shows the execution plan (what will change).
5. 'terraform apply' provisions the actual resources.
6. Terraform saves the state of resources in 'terraform.tfstate'.
7. Outputs display important information like public IPs or IDs.
8. 'terraform destroy' removes all created resources.

Concept Map Overview

Terraform Workflow (Simplified):

```
 [ main.tf (Code) ]
          |
          v
   [ Provider (AWS) ]
          |
          v
   [ Terraform Plan ]
          |
          v
   [ Apply -> AWS Cloud ]
          |
          v
   [ State File (.tfstate) ]
          |
          v
   [ Outputs / Logs ]
```

Summary

Terraform enables consistent, repeatable infrastructure automation across multiple cloud platforms. By mastering key concepts like Providers, Resources, State, and Modules, you can automate cloud deployments efficiently and securely.