Terraform State Commands

HashiCorp Terraform

State Commands

- Terraform State is automatically managed by Terraform
    - Terraform provides commands to safely work with the state file, for reading and editing
- Terraform state commands allows you to perform various operational tasks:
    - Inspecting State
    - Manipulating State
    - Advanced troubleshooting
- Common usecases:
    - Moving resources around (Advanced)
    - Removing resources from the state file (Advanced)
    - Importing resources into the state file
    - Viewing list of managed resources in current configuration
- Changes to the State file does not affect resources in the real world
    - Alters how Terraform views the resources internally

HashiCorp Terraform

Inspecting State

- Inspecting state file is useful for getting overview of all resources created, or retrieving information about a specific resource
- Common usecases:
  - Getting list of all resources created in particular infrastructure
  - Retrieving the instance id of an EC2 instance resource
  - Getting a list of all resources created, then filtering by keywords
- Inspecting state file is a read only operation - does not alter the state file
- terraform state list
  - Shows all resources under current configuration
- terraform state show <resource_address>
  - Show details about specific resource
  - terraform state show aws_instance.main

Manipulating State

- Common usecases:
  - Moving resources around for specific ordering
  - Renaming resources due to a change of module or resource names
  - Removing resources from state file as they no longer exist
- terraform state mv <source resource_address> <destination resource_address>
  - Moves a resource to a different resource address
- terraform state rm <resource_address>
  - Removes a resource from your state file
- Manipulating state file does not directly change infrastructure in real world
  - But side affects can lead to proposed changes

Advanced Troubleshooting

- Very rare case - you may have to directly edit state file
  - Rarely happens when certain attributes are no longer required, and Terraform failed to clean up
  - Or dependency lost its tracking and have to be removed
- You may directly edit state file by pulling state file down
  - State files can be pulled down into json file
  - After changes are made, they can be pushed back up into their remote backend
- terraform state pull
  - Pulls your state file down from backend
  - Common to use redirect, terraform state pull > state.json
- terraform state push <PATH>
  - Pushes the state file back to the backend
  - <PATH> is the location of the state file you wish to push back up
- You may also pull down the state file, to view criticial metadata about infrastructure
  - Which Terraform version is being used?

HashiCorp
Terraform