Notes

• `terraform validate`: Checks the syntax of the Terraform files and verifies that they are internally consistent, but does not ensure that the resources exist or that the providers are properly configured.
• `terraform fmt`: Automatically updates Terraform configuration files to a canonical format and style, improving consistency and readability. The command works only for the files in the current working directory, but you can also add a `-recursive` flag to format `.tf` files in nested directories.
• `terraform plan`: Creates an execution plan, showing what actions Terraform will take to achieve the desired state defined in the Terraform files. This command does not modify the actual resources or state.
• `terraform plan -out <filename>`: Similar to `terraform plan`, but it also writes the execution plan to a file that can be used by `terraform apply`, ensuring that exactly the planned actions are taken.
• `terraform apply`: Applies the execution plan, making the necessary changes to reach the desired state of the resources. If you run `terraform plan` with the `-out` option, you can run `terraform apply <filename>` to provide the execution plan.
• `terraform show`: Provides human-readable output from a state or plan file. It's used to inspect the current state or to see the actions planned by a `terraform plan` command.
• `terraform state list`: Lists all resources in the state file, useful for managing and manipulating the state.
• `terraform destroy`: Destroys all resources tracked in the state file. This command is the equivalent of passing a `-destroy` flag to the `terraform apply` command.
• `terraform -help`: Provides help information about Terraform commands. It can be used alone for a general overview, or appended to a specific command for detailed help about that command.

```
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  metadata      Metadata related commands
  modules       Show all declared modules in a working directory
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  stacks        Manage HCP Terraform stack operations
  state         Advanced state management
  taint         Mark a resource instance as not fully functional
  test          Execute integration tests for Terraform modules
  untaint       Remove the 'tainted' state from a resource instance
  version       Show the current Terraform version
  workspace     Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR    Switch to a different working directory before executing the
                given subcommand.
  -help         Show this help output or the help for a specified subcommand.
  -version      An alias for the "version" subcommand.

  -input=true            Ask for input for variables if not directly set.

  -no-color              If specified, output won't contain any color.

  -parallelism=n         Limit the number of parallel resource operations.
                         Defaults to 10.

  -replace=resource      Terraform will plan to replace this resource instance
                         instead of doing an update or no-op action.

  -state=path            Path to read and save state (unless state-out
                         is specified). Defaults to "terraform.tfstate".

  -state-out=path        Path to write state to that is different than
                         "-state". This can be used to preserve the old
                         state.

  -var 'foo=bar'         Set a value for one of the input variables in the root
                         module of the configuration. Use this option more than
                         once to set more than one variable.

  -var-file=filename     Load variable values from the given file, in addition
                         to the default files terraform.tfvars and *.auto.tfvars.
                         Use this option more than once to include more than one
                         variables file.

  If you don't provide a saved plan file then this command will also accept
  all of the plan-customization options accepted by the terraform plan command.
  For more information on those options, run:
      terraform plan -hel
```