

quick cheat sheet for Terraform:

1. Terraform Basics:

- **terraform init**: initializes a working directory containing Terraform configuration files
- **terraform plan**: shows what changes Terraform will make before actually executing them
- **terraform apply**: executes the changes described in the Terraform configuration files
- **terraform destroy**: destroys the infrastructure created by Terraform

2. Terraform Configuration:

- **.tf** files: Terraform configuration files
- **provider** block: specifies the cloud provider and the credentials to access it
- **resource** block: defines the infrastructure components to create, modify, or destroy
- **variable** block: declares input variables that are used in the Terraform configuration
- **output** block: declares output variables that can be used by other Terraform configurations or scripts

3. Terraform Commands:

- **terraform show**: shows the current state of the infrastructure
- **terraform state**: shows the current state of a specific resource
- **terraform refresh**: updates the Terraform state file with the current state of the infrastructure
- **terraform import**: imports an existing resource into the Terraform state file

4. Terraform Modules:

- **module** block: defines a reusable component of the Terraform configuration
- **source** parameter: specifies the location of the module, either a local path or a remote repository
- **output** block: declares output variables that are exposed by the module
- **terraform get**: downloads modules from the specified sources
- **terraform init -upgrade**: upgrades the modules to the latest version

5. Terraform State:

- **terraform state pull**: downloads the current state of the infrastructure from the remote state backend
- **terraform state push**: uploads the current state of the infrastructure to the remote state backend
- **terraform state list**: lists all the resources in the Terraform state file
- **terraform state mv**: moves a resource to a new name or a new module
- **terraform state rm**: removes a resource from the Terraform state file

Advanced level Terraform concepts and commands that you might find useful:

Backend configuration: Terraform provides different backend types like S3, Consul, etcd, and more. You can configure the backend using the backend block in the configuration file.

Resource Targeting: You can apply changes to specific resources by using the -target flag followed by the resource address. This is useful when you only want to apply changes to a specific resource instead of the entire infrastructure.

Terraform Modules: Modules help you reuse the code and make it easy to maintain. You can create a module with specific functionalities and reuse it in different Terraform projects.

Terraform Workspaces: Workspaces allow you to manage multiple environments like staging, production, and development within a single Terraform configuration. You can switch between different workspaces using the `terraform workspace select` command.

Terraform Providers: Terraform Providers are plugins that allow you to interact with different cloud providers. You can find providers for different cloud providers like AWS, GCP, Azure, and more.

Terraform Functions: Functions in Terraform are used to manipulate data and perform different operations. There are different types of functions available like string functions, list functions, map functions, and more.

Terraform Variables: Variables are used to parameterize the Terraform configuration. You can define variables in the configuration file or pass them through the command line using the `-var` flag.

Terraform State Management: Terraform state file is used to store the current state of the infrastructure. You can configure the state file using the `terraform.tfstate` file or by using a remote backend.

Terraform Outputs: Outputs allow you to extract data from the Terraform state and use it in other parts of the infrastructure. You can define outputs using the `output` block in the configuration file.

Terraform Graph: The `terraform graph` command allows you to visualize the infrastructure dependencies graphically.

These are just a few advanced level Terraform concepts and commands. Terraform is a powerful tool with many features, and it's always a good idea to read the official documentation to learn more.