

# Devops - III

Sud

---

---

---

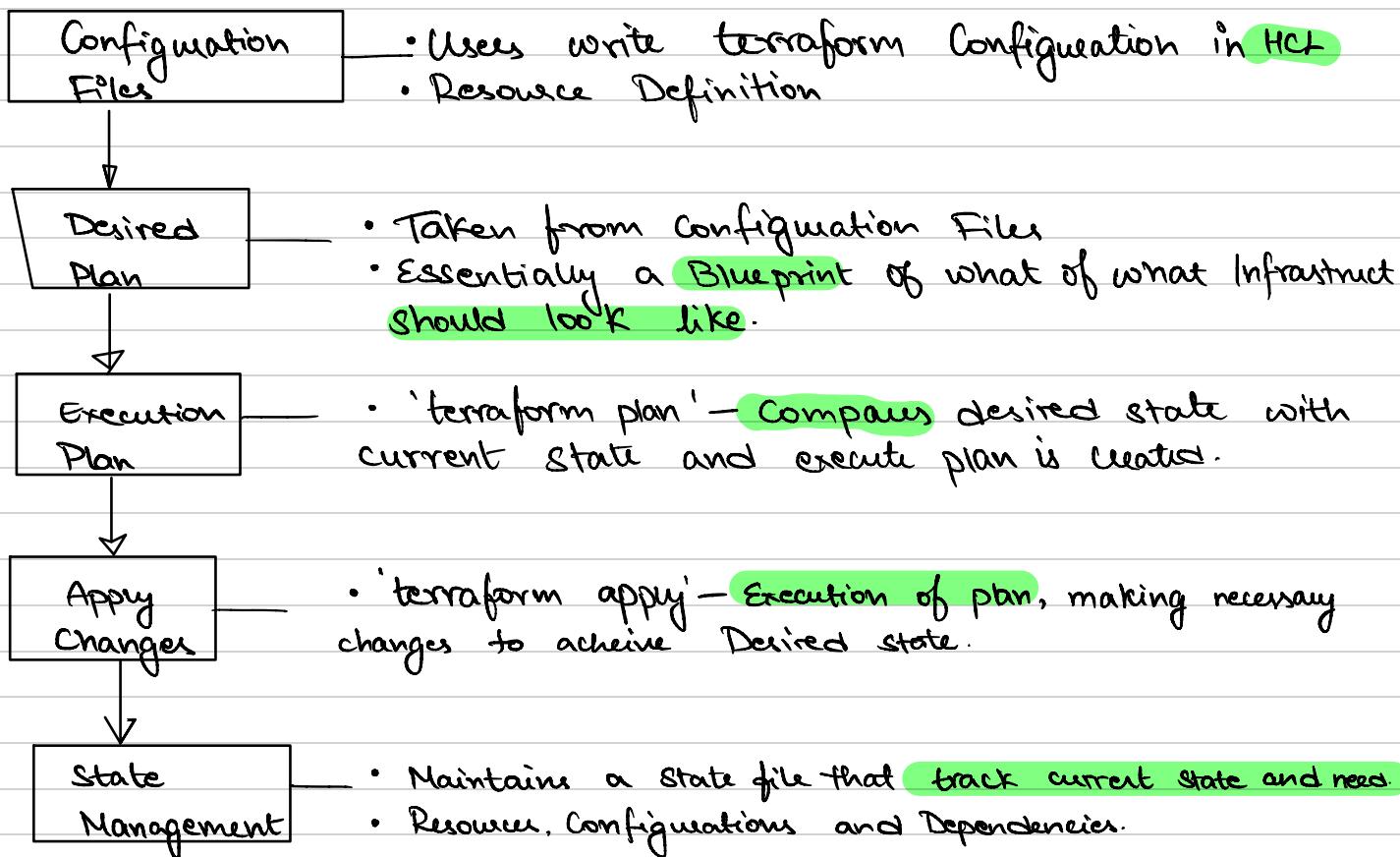


# TERRAFORM

→ Declarative tool

- \* It is a powerful, Open-Source, Infrastructure as a Code tool.
- \* Developed by Hashicorp.
- \* Enables user to describe and provision infrastructure resource in Declarative manner
  - This means that instead of writing scripts, we need to give what desired state is needed and Terraform performs it automatically.

## How does Terraform Work? (not as in ppt)



## KEY features: (DRMSSM)

- #1 - Declarative Configuration - Terraform uses declarative syntax rather than specifying
- #2 - Resource Provisioning - User define Resources and Configuration in HCL script.
- #3 - Multi-Cloud Support - Terraform is cloud-agnostic & support multiple cloud provider.
- #4 - State Management - Terraform maintains a state file that tracks current state.
  - allow user to break down infrastructure code into reusable module
- #5 - Modularity & Reusability: Terraform configurations are modular. This promotes Code Organization, collaboration, standardization.

# TERRAFORM COMMANDS (main.tf file)

INIT

PLAN

APPLY

DESTROY

## INIT:

- \* Initialize Working Directory containing Configuration Files
  - Initialise Back-End
  - Download Provider plugin
  - Download Modules
  - Create local state file.

## PLAN:

- \* Preview changes terraform will make on infrastructure based on current configuration & desired state.
  - Adds Resources
  - Modify code to attain Desired state.
  - Delete - no longer used files.

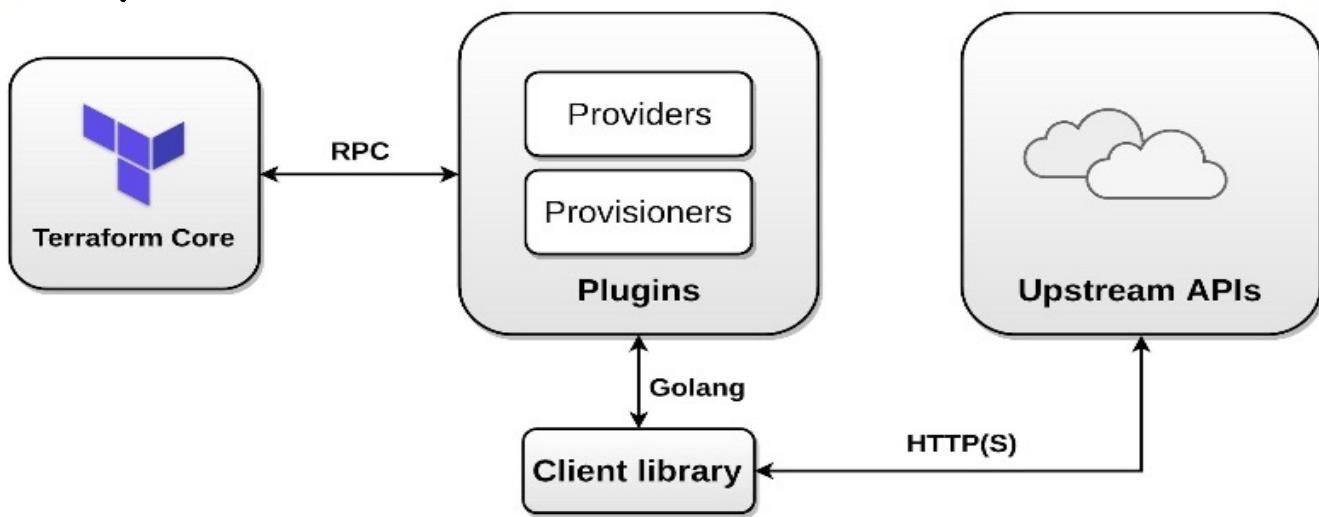
## APPLY:

- \* Apply changes from preview to the actual infrastructure.
  - Execute changes
  - Confirmation prompt

## DESTROY:

- \* Used to destroy all resources created by particular Terraform configuration.
  - Identify Resources.
  - Delete Resources
  - Confirmation prompt.

## Infrastructure:

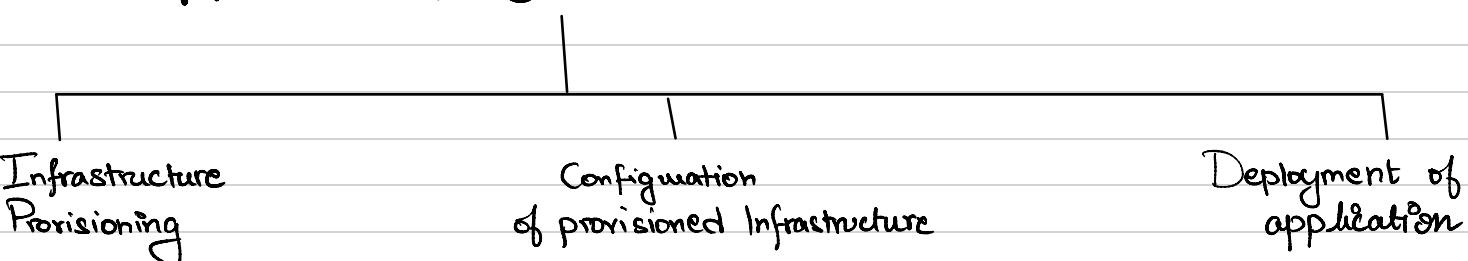


# Infrastructure as a Code

Mainly used to Automate tasks

- \* IaC enables management of Infrastructure through Code and automation.
- \* This Revolutionizes how organizations build, deploy & Manage their Infrastructure bringing agility, Scalability and Consistency to process.

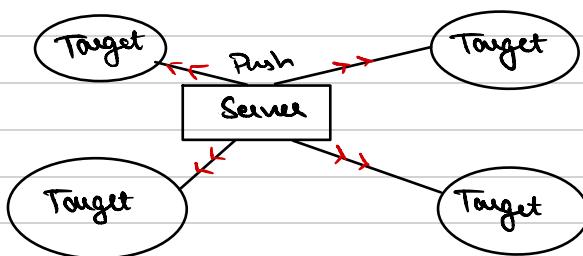
## MAIN TASK CATEGORIES OF IAAC



## PUSH / PULL APPROACH IN IAAC

### Push:

- \* Central Management Server initiates and controls Deployment process.
- \* Pushes Configuration changes (or) updates to target.
- \* Centralized Approach.



### Pull:

- \* Infrastructure components periodically pulling Configuration update from a central Server
- \* De-centralized Approach.

## PLUGIN BASED ARCHITECTURE

- \* Allows for modularity and extensibility
- \* Allows Integration of 3rd party plugins for seamless experience / functionality
- \* Promotes flexibility & Adaptability.
- \* Terraform, Ansible, Chef, Jenkins

# ANSIBLE

- \* It is an Open-Source automation tool for configuration management, application deployment and task automation.
- \* Developed by Red-Hat.
- \* Ansible simplifies complex tasks by allowing users to define Iaas using human-readable YAML files.

## COMPONENTS OF ANSIBLE (PIMRH)

#1 PLAYBOOKS : • These are YAML files where user defines a series of tasks to be executed on managed node.

#2 INVENTORY : • Files or directories that list managed nodes and/or their grouping.

#3 MODULES : • Unit of work that Ansible executes  
• Perform specific tasks like installing packages, managing services and/or handling files.

#4 ROLES: • Roles are a way to group related tasks, variables, files and templates.  
• Makes tasks more modular and maintainable.

#5 HANDLER: • Handlers are tasks that are triggered by other tasks.  
• Typically used to restart services or perform cleanup actions.

## KEY-FEATURES

- \* Agentless-architecture: Ansible operates w/o the need for Agents
- \* Simple YAML syntax: Ansible are operated on simple, human-readable YAML.
- \* Idempotent: Repeated executions of playbooks will not cause unintended changes.
- \* Extensible modules: Has wide range of built-in modules that can manage various aspect of IT environment.
- \* Declarative and Procedural: Provides flexibility in providing both declarative and procedural towards task automation.