Terraform Task

**Name:Shaik Khaja Basha**

**Batch : Batch 11**

**Date : 24.07.2025**

**Task : ignore canges & replace triggred**

1. **Explore the life cycle of ignore\_changes and replace\_triggered\_by** ?

**Ans:** In Terraform, ignore\_changes and replace\_triggered\_by are lifecycle meta-arguments that control how resources are managed. ignore\_changes prevents Terraform from attempting to update specific attributes that are being modified outside of Terraform, while replace\_triggered\_by forces a resource to be replaced when specific referenced resources or variables change

**ignore\_changes:**

**purpose:**

* To ignore changes to specific attributes of a resource that are being managed outside of Terraform.
* Even if instance\_type changes in code or tfvars, Terraform will **not trigger a replacement** or update the resource.

**Use Cases:**

* + When an external process (like an Azure Policy) modifies resource attributes, and Terraform should not attempt to revert those changes.
  + When dealing with dynamic data that may change after resource creation, but should not trigger a resource replacement.

**Example:**

Ignoring changes to tags on an AWS instance that are being managed by an external tagging process.

**working:**

Terraform will still manage the resource's creation and destruction, but it will not attempt to reconcile the specified attributes with the Terraform configuration.

**Syntax:**

resource "aws\_instance" "example" {

ami = "ami-123456"

instance\_type = var.instance\_type

lifecycle {

ignore\_changes = [instance\_type]

}

}

**Scenario**:  
You manage an Azure App Service. Operations team might change settings manually, and you don’t want Terraform to revert their changes every time.

**Example:**

lifecycle {

ignore\_changes = [ app\_settings ]

}

Terraform will not re-apply changes if only app\_settings are different in the state vs config.

**Effect:**

* Terraform won't recreate the resource if only the timestamp changes
* Useful when you want to keep the resource stable despite volatile attributes

**replace\_triggered\_by:**

**Purpose:**

To force a resource to be recreated when specific referenced resources or variables change

**Use Cases:**

* + When a resource depends on another resource whose changes should trigger a full replacement of the dependent resource.
  + When a resource needs to be replaced when a specific variable or attribute of another resource changes.
  + If the local\_file.example resource is modified or recreated, null\_resource.example will also be replaced.

**Example:**

Replacing a Google Compute Instance when its startup script changes.

**How it works:**

When the referenced resource or variable changes, Terraform will plan a replacement of the resource, effectively destroying and recreating it.

**Syntax:**

resource "null\_resource" "example" {

triggers = {

value = var.value

}

lifecycle {

replace\_triggered\_by = [

var.value,

local\_file.example # or another resource

]

}

}

**Example with Random Provider:**

resource "random\_id" "seed" {

byte\_length = 4

}

resource "null\_resource" "dependent" {

triggers = {

seed = random\_id.seed.hex

}

lifecycle {

replace\_triggered\_by = [random\_id.seed] *# Recreate when seed changes*

}

}

**Effect:**

* null\_resource.dependent will be destroyed/recreated whenever random\_id.seed changes
* Creates an explicit dependency link

Difference between ignore\_changes and replace\_triggered\_by

|  |  |
| --- | --- |
| **ignore\_changes** | **replace\_triggered\_by** |
| ignore\_changes prevents updates to specific attributes | replace\_triggered\_by forces a full resource replacement. |
| ignore\_changes is useful for managing resources where external processes make changes | replace\_triggered\_by is useful for managing resources that depend on other resources that may change. |
| ignore\_changes allows for shared management of a resource between Terraform and external processes | replace\_triggered\_by ensures consistency by replacing the resource when a specific dependency changes |