Data Engineering 101

Terraform

Only the essentials required by a Data Engineer

Shwetank Singh
GritSetGrow - GSGLearn.com

Providers

PROVIDERS ARE RESPONSIBLE FOR UNDERSTANDING API INTERACTIONS AND EXPOSING RESOURCES. EACH PROVIDER HAS ITS OWN SET OF RESOURCE TYPES.

```
terraform {
    required_providers
        { aws = { source =
"hashicorp/aws" version = "~> 3.0" }}}
provider "aws" { region = "us-west-2" }
```

DEFINES THE AWS PROVIDER AND SPECIFIES THAT IT SHOULD USE THE US-WEST-2 REGION.

Shwetank Singh
GritSetGrow - GSGLearn.com

Resources

A RESOURCE DEFINES A COMPONENT IN YOUR INFRASTRUCTURE, LIKE AN EC2 INSTANCE OR S3 BUCKET.

```
resource "aws_instance" "example"
{ ami = "ami-123456"
instance_type = "t2.micro" }
```

CREATES AN AWS EC2 INSTANCE WITH THE SPECIFIED AMI AND INSTANCE TYPE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Data Sources

DATA SOURCES ALLOW YOU TO RETRIEVE INFORMATION ABOUT EXISTING RESOURCES OUTSIDE TERRAFORM'S MANAGEMENT.

```
data "aws_ami" "example"
{ most_recent = true
  owners = ["self"] }
```

RETRIEVES THE MOST RECENT AMI OWNED BY THE CURRENT USER.

Shwetank Singh
GritSetGrow - GSGLearn.com

Variables

VARIABLES ARE USED TO CUSTOMIZE TERRAFORM CONFIGURATIONS WITHOUT HARDCODING VALUES.

```
variable "instance_type"
{ default = "t2.micro" }
```

DEFINES A VARIABLE INSTANCE_TYPE WITH A DEFAULT VALUE OF T2.MICRO.

Shwetank Singh
GritSetGrow - GSGLearn.com

Output

OUTPUTS DISPLAY VALUES AFTER A TERRAFORM RUN, SUCH AS RESOURCE IDS.

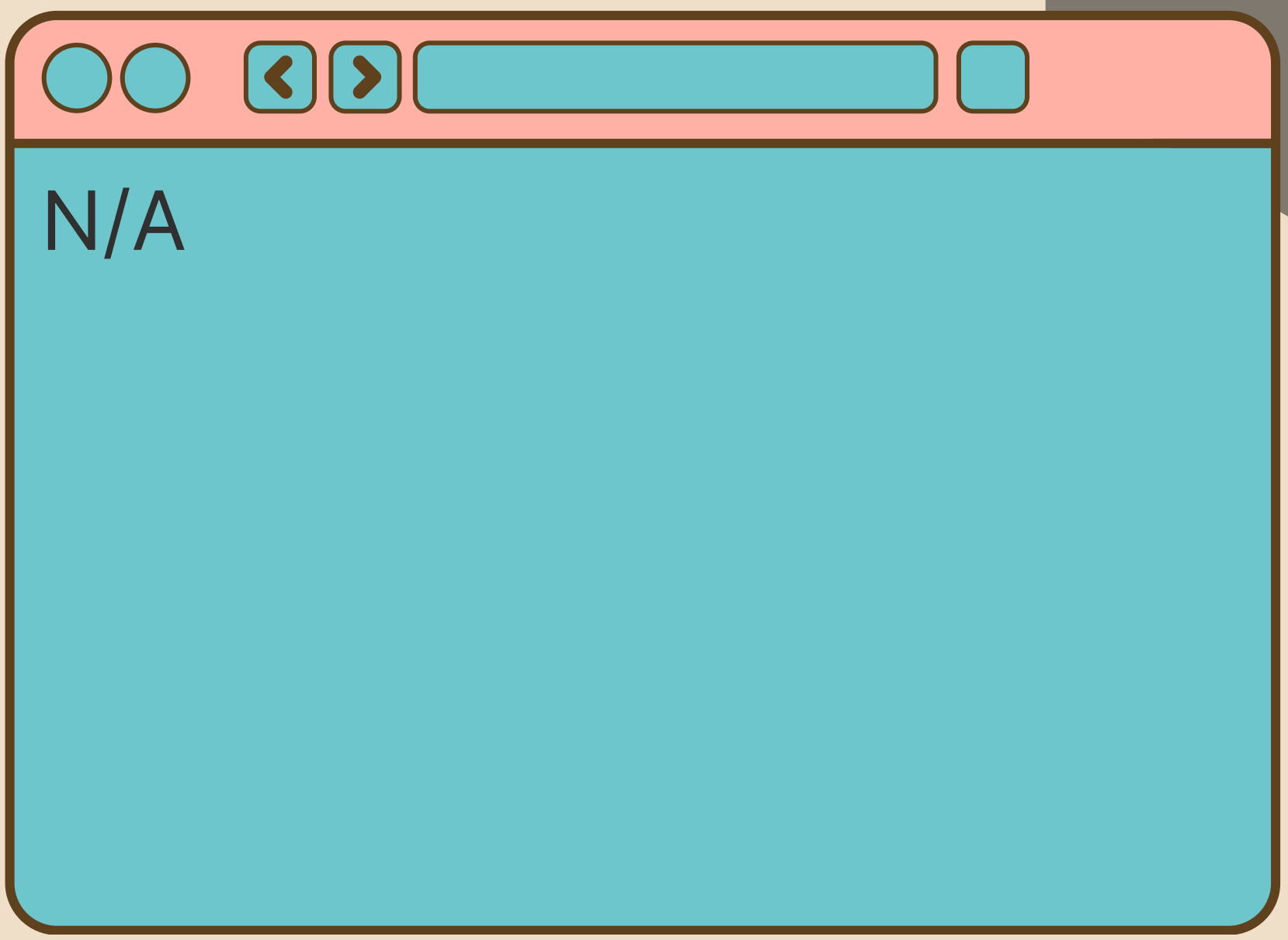
```
output "instance_ip"
{
value = aws_instance.example.public_ip
}
```

DISPLAYS THE PUBLIC IP ADDRESS OF AN EC2 INSTANCE AFTER DEPLOYMENT.

Shwetank Singh
GritSetGrow - GSGLearn.com

State File

THE STATE FILE TRACKS THE RESOURCES TERRAFORM MANAGES. IT ALLOWS TERRAFORM TO KNOW WHAT RESOURCES IT CONTROLS.



TERRAFORM USES THE STATE FILE TO MAP REAL-WORLD RESOURCES TO YOUR CONFIGURATION.

Shwetank Singh
GritSetGrow - GSGLearn.com

Modules

MODULES ARE CONTAINERS FOR MULTIPLE RESOURCES USED TOGETHER. THEY PROMOTE REUSABILITY OF CODE.

```
module "vpc" {
source = "terraform-aws-modules/vpc/aws"
version = "3.0.0" }
```

THE VPC MODULE IS CALLED FROM THE TERRAFORM REGISTRY, SIMPLIFYING THE CREATION OF A VPC.

Shwetank Singh
GritSetGrow - GSGLearn.com

Provisioners

PROVISIONERS EXECUTE SCRIPTS OR COMMANDS AFTER A RESOURCE IS CREATED OR DESTROYED.

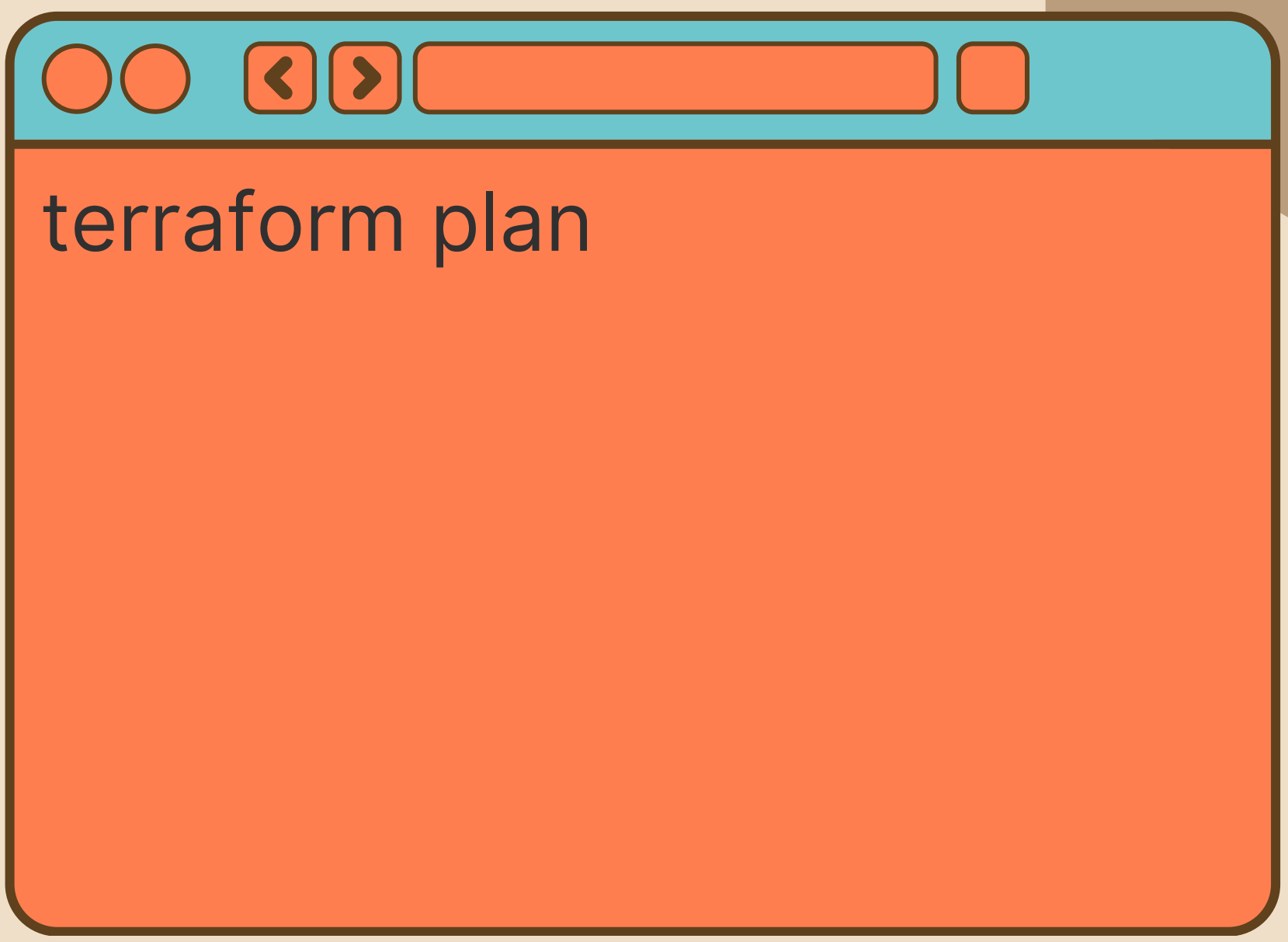
```
resource "aws_instance" "example"
{ ami = "ami-123456"
instance_type = "t2.micro"
provisioner "local-exec"
    { command = "echo
${aws_instance.example.public_ip}" } }
```

EXECUTES A LOCAL COMMAND TO PRINT THE INSTANCE'S PUBLIC IP ADDRESS AFTER CREATION.

Shwetank Singh
GritSetGrow - GSGLearn.com

Terraform Plan

TERRAFORM PLAN SHOWS THE CHANGES THAT WILL BE MADE WITHOUT ACTUALLY APPLYING THEM.



GENERATES AND DISPLAYS THE EXECUTION PLAN WITHOUT MAKING CHANGES.

Shwetank Singh
GritSetGrow - GSGLearn.com

Terraform Apply

TERRAFORM APPLY APPLIES THE CHANGES REQUIRED TO REACH THE DESIRED STATE OF THE CONFIGURATION.

```
terraform apply
```

APPLIES THE CONFIGURATION CHANGES TO REACH THE DESIRED STATE IN YOUR INFRASTRUCTURE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Backend

BACKENDS DETERMINE HOW STATE IS LOADED AND HOW OPERATIONS ARE EXECUTED.

```
terraform {backend "s3"
{ bucket = "my-tf-state"
key = "global/s3/terraform.tfstate"
region = "us-west-2" } }
```

CONFIGURES S3 AS THE BACKEND TO STORE THE STATE FILE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Remote State

REMOTE STATE ALLOWS TERRAFORM TO STORE STATE FILES IN A REMOTE STORAGE LOCATION LIKE AWS S3, ENSURING MULTIPLE USERS CAN ACCESS IT.

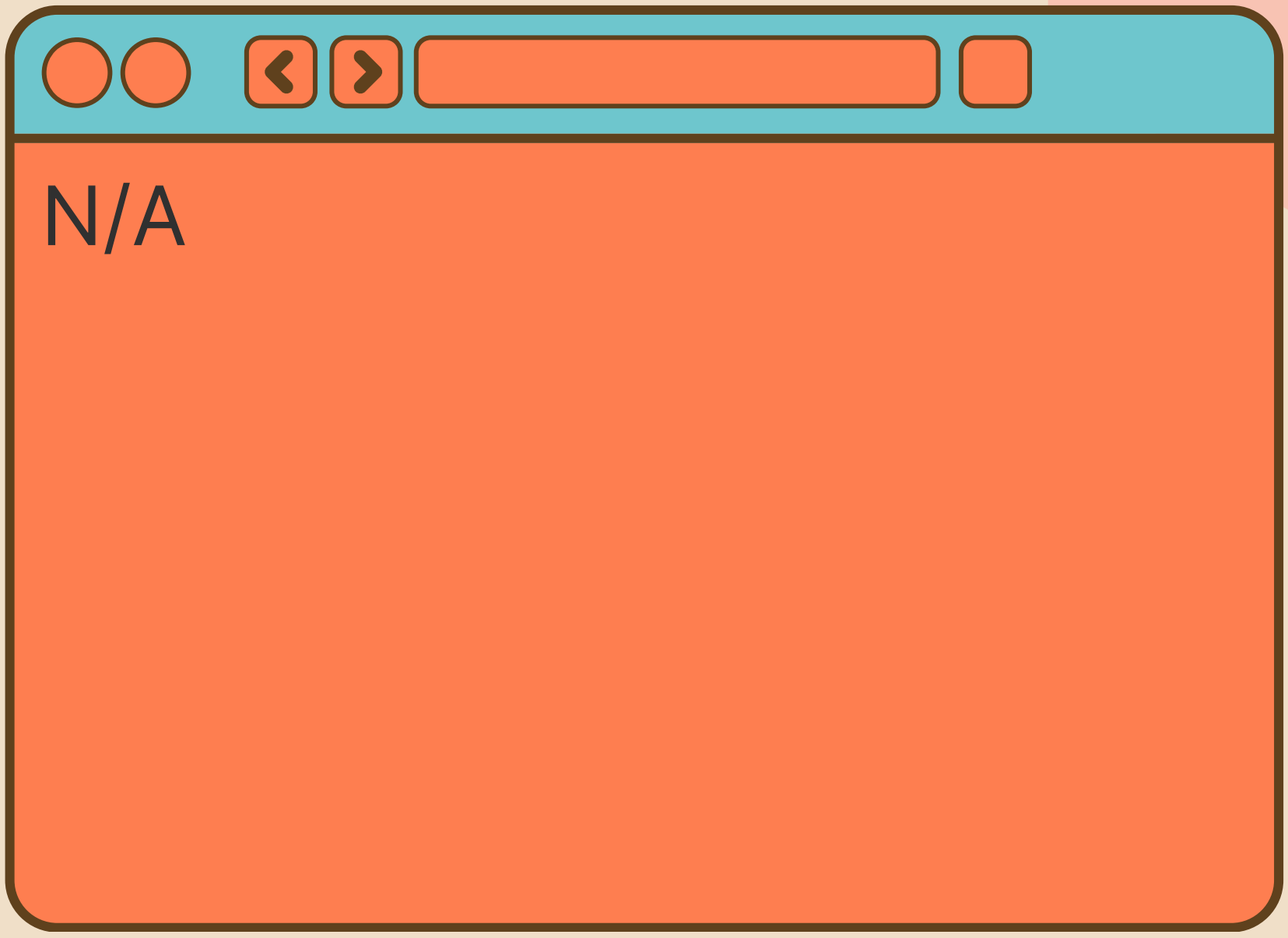
```
terraform { backend "s3" { bucket =
"mybucket" key = "path/to/my/key"
region = "us-west-1" } }
```

STORES THE STATE FILE IN AN S3 BUCKET FOR REMOTE MANAGEMENT.

Shwetank Singh
GritSetGrow - GSGLearn.com

Locking State

STATE LOCKING ENSURES THAT ONLY ONE TERRAFORM PROCESS MODIFIES THE STATE AT A TIME, PREVENTING CONFLICTS.



TERRAFORM AUTOMATICALLY LOCKS STATE WHEN MODIFYING IT USING BACKENDS THAT SUPPORT LOCKING, SUCH AS DYNAMODB IN AWS.

Shwetank Singh
GritSetGrow - GSGLearn.com

Terraform Destroy

TERRAFORM DESTROY REMOVES ALL RESOURCES DEFINED IN THE CONFIGURATION.

```
terraform destroy
```

DESTROYS ALL RESOURCES DEFINED IN THE TERRAFORM CONFIGURATION.

Shwetank Singh
GritSetGrow - GSGLearn.com

Workspaces

WORKSPACES ALLOW YOU TO MAINTAIN MULTIPLE STATE FILES AND SEPARATE ENVIRONMENTS LIKE DEVELOPMENT AND PRODUCTION.

```
terraform workspace new dev
```

CREATES A NEW WORKSPACE CALLED DEV, WHICH MAINTAINS A SEPARATE STATE FILE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Meta-Arguments

META-ARGUMENTS LIKE DEPENDS_ON, COUNT, AND FOR_EACH CUSTOMIZE RESOURCE BEHAVIOR.

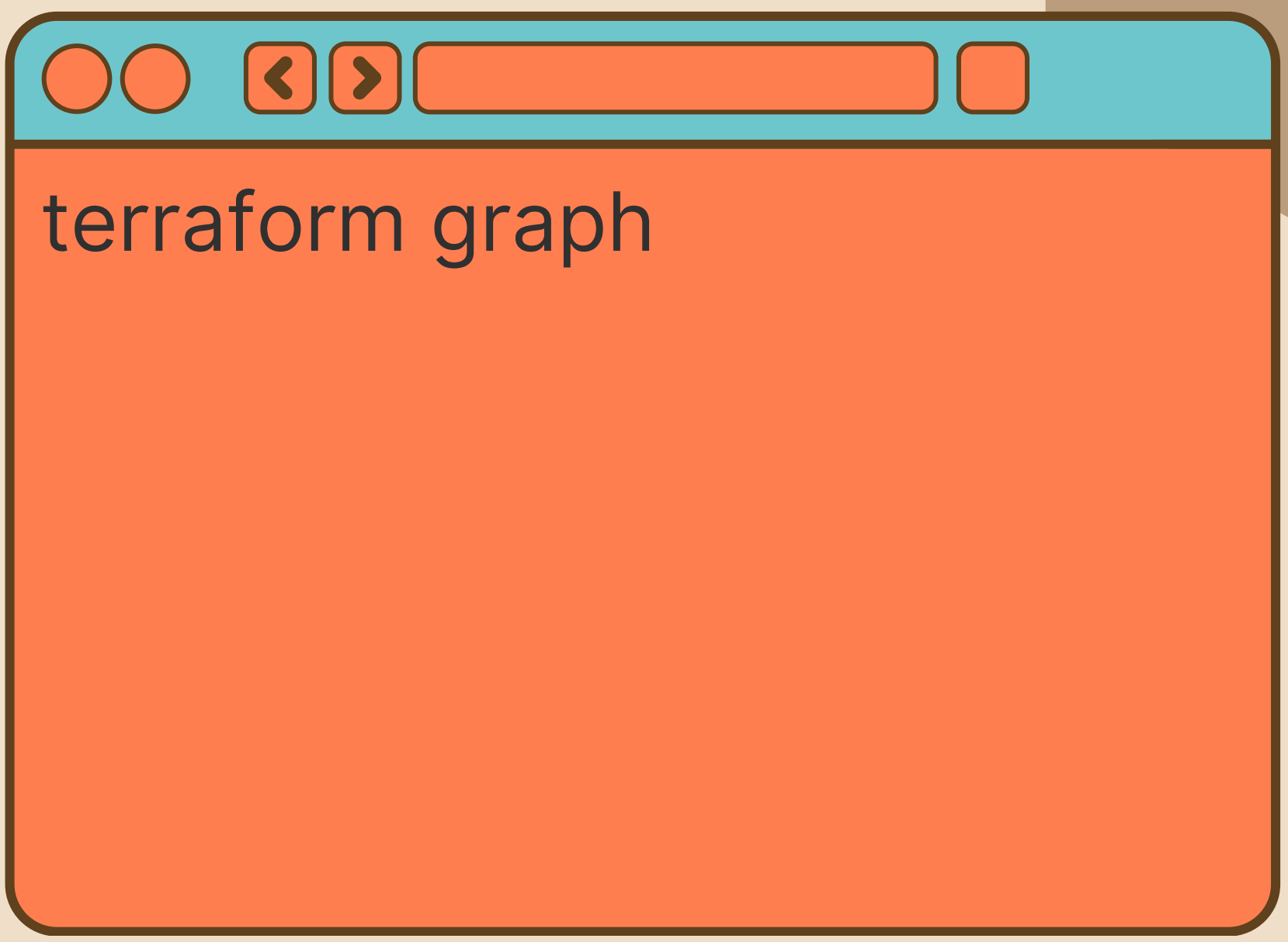
```
resource "aws_instance" "example"
{ count = 2
ami = "ami-123456"
instance_type = "t2.micro" }
```

CREATES TWO EC2 INSTANCES USING THE COUNT ARGUMENT.

Shwetank Singh
GritSetGrow - GSGLearn.com

Terraform Graph

TERRAFORM GRAPH GENERATES A VISUAL REPRESENTATION OF THE TERRAFORM RESOURCES AND THEIR RELATIONSHIPS.



CREATES A DOT FORMAT GRAPH OF THE INFRASTRUCTURE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Terraform Import

TERRAFORM IMPORT IMPORTS EXISTING RESOURCES INTO TERRAFORM MANAGEMENT.

```
terraform import aws_instance.example
i-1234567890abcdef0
```

IMPORTS AN EXISTING EC2 INSTANCE INTO TERRAFORM STATE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Outputs and Dependencies

OUTPUTS CAN BE REFERENCED IN OTHER TERRAFORM CONFIGURATIONS, ESTABLISHING DEPENDENCIES BETWEEN RESOURCES.

```
output "vpc_id" {
value = aws_vpc.example.id }
```

OUTPUTS THE VPC ID, WHICH CAN BE REFERENCED BY OTHER CONFIGURATIONS.

Shwetank Singh
GritSetGrow - GSGLearn.com

Conditional Expressions

CONDITIONAL EXPRESSIONS ALLOW TERRAFORM TO DYNAMICALLY CONFIGURE RESOURCES BASED ON SPECIFIC CONDITIONS.

```
resource "aws_instance" "example"
{ instance_type = var.instance_size
== "large" ? "t2.large" : "t2.micro" }
```

CREATES AN EC2 INSTANCE BASED ON THE VARIABLE INSTANCE_SIZE VALUE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Tainting Resources

TAINTING FORCES TERRAFORM TO DESTROY AND RECREATE A RESOURCE ON THE NEXT APPLY.

```
terraform taint aws_instance.example
```

MARKS THE EC2 INSTANCE AS TAINTED, FORCING IT TO BE RECREATED.

Shwetank Singh
GritSetGrow - GSGLearn.com

Resource Dependencies

TERRAFORM AUTOMATICALLY CREATES RESOURCE DEPENDENCIES USING EXPLICIT REFERENCES BETWEEN RESOURCES.

```
resource "aws_instance" "example"
{ ami = "ami-123456"
subnet_id = aws_subnet.example.id
}
```

THE EC2 INSTANCE DEPENDS ON THE SUBNET BEING CREATED FIRST.

Shwetank Singh
GritSetGrow - GSGLearn.com

Sensitive Values

SENSITIVE VALUES MASK OUTPUT AND PREVENT THEM FROM BEING DISPLAYED.

```
variable "db_password"
{ type = string
  sensitive = true
}
```

MARKS THE DATABASE PASSWORD AS SENSITIVE, PREVENTING IT FROM BEING PRINTED.

Shwetank Singh
GritSetGrow - GSGLearn.com

Terragrunt

TERRAGRUNT IS A WRAPPER FOR TERRAFORM THAT PROVIDES EXTRA TOOLS FOR MANAGING MULTIPLE TERRAFORM CONFIGURATIONS.

```
terragrunt {
terraform {
source = "../modules/my-module"
} }
```

USES TERRAGRUNT TO CALL TERRAFORM MODULES FROM A SOURCE DIRECTORY.

Shwetank Singh
GritSetGrow - GSGLearn.com

Terraform Registry

TERRAFORM REGISTRY IS A PUBLIC COLLECTION OF REUSABLE TERRAFORM MODULES.

```
module "vpc"
{ source = "terraform-aws-modules/vpc/aws"
version = "2.0.0" }
```

CALLS A VPC MODULE FROM THE TERRAFORM REGISTRY.

Shwetank Singh
GritSetGrow - GSGLearn.com

Managed and Unmanaged Resources

Managed resources are created and managed by Terraform, while unmanaged resources exist outside of Terraform's control.

```
terraform import
aws_s3_bucket.example existing-bucket
```

Imports an unmanaged S3 bucket into Terraform's state file to be managed.

Shwetank Singh
GritSetGrow - GSGLearn.com

Dynamic Blocks

DYNAMIC BLOCKS ALLOW YOU TO GENERATE MULTIPLE INSTANCES OF A NESTED BLOCK INSIDE A RESOURCE.

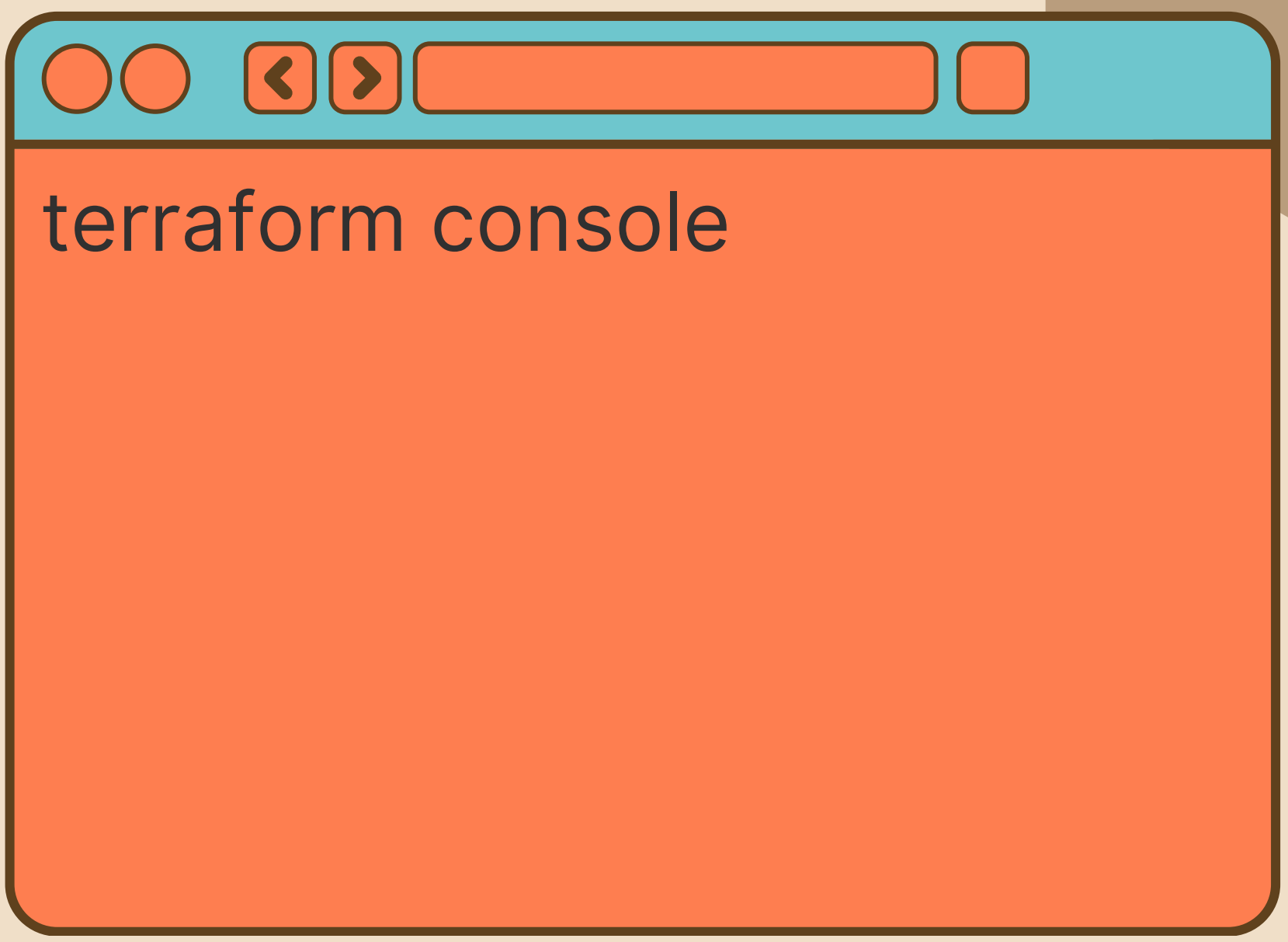
```
resource "aws_security_group" "example"
{ dynamic "ingress"
{ for_each = var.allowed_ports content
{ from_port = ingress.value
to_port = ingress.value
protocol = "tcp"
} } }
```

DYNAMICALLY CREATES MULTIPLE INGRESS RULES FOR A SECURITY GROUP.

Shwetank Singh
GritSetGrow - GSGLearn.com

Terraform Console

TERRAFORM CONSOLE OPENS AN INTERACTIVE CONSOLE FOR EVALUATING EXPRESSIONS WITHIN THE CONFIGURATION.



OPENS THE CONSOLE TO TEST EXPRESSIONS, VARIABLES, AND INTERPOLATIONS.

Shwetank Singh
GritSetGrow - GSGLearn.com

Terraform Refresh

TERRAFORM REFRESH UPDATES THE STATE FILE WITH THE CURRENT RESOURCE STATE WITHOUT MAKING CHANGES.

```
terraform refresh
```

UPDATES TERRAFORM STATE TO REFLECT THE REAL-WORLD STATE OF RESOURCES.

Shwetank Singh
GritSetGrow - GSGLearn.com

Parallelism

TERRAFORM APPLIES RESOURCES IN PARALLEL, BUT YOU CAN LIMIT THIS USING THE -PARALLELISM FLAG.

```
terraform apply -parallelism=5
```

LIMITS THE NUMBER OF CONCURRENT OPERATIONS TO 5 DURING THE APPLY PHASE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Local Values

LOCAL VALUES ARE LIKE VARIABLES BUT ARE SCOPED TO THE CURRENT MODULE, ALLOWING FOR CLEANER AND MORE MAINTAINABLE CODE.

```
locals { instance_size = "t2.micro" }
```

DEFINES A LOCAL VALUE INSTANCE_SIZE THAT CAN BE USED ELSEWHERE IN THE MODULE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Iteration with for_each

FOR_EACH ALLOWS ITERATION OVER A SET OF ITEMS, CREATING MULTIPLE INSTANCES OF A RESOURCE.

```
resource "aws_instance" "example"
{ for_each = toset(var.instance_names)
instance_type = "t2.micro"
ami = "ami-123456" }
```

CREATES AN EC2 INSTANCE FOR EACH ITEM IN THE INSTANCE_NAMES VARIABLE.

Shwetank Singh
GritSetGrow - GSGLearn.com

External Data Sources

EXTERNAL DATA SOURCES ALLOW TERRAFORM TO FETCH DATA FROM OUTSIDE SOURCES LIKE SCRIPTS.

```
data "external" "example"
{ program = ["bash", "./get_data.sh"]
}
```

EXECUTES AN EXTERNAL SCRIPT TO RETRIEVE DATA FOR USE IN THE CONFIGURATION.

Shwetank Singh
GritSetGrow - GSGLearn.com

Timeouts

SOME RESOURCES SUPPORT TIMEOUTS TO HANDLE CASES WHERE OPERATIONS TAKE LONGER THAN EXPECTED.

```
resource "aws_instance" "example"
{ ami = "ami-123456"
instance_type = "t2.micro"
timeouts { create = "10m"
           delete = "15m" } }
```

CONFIGURES TIMEOUTS FOR THE CREATION AND DELETION OF AN EC2 INSTANCE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Custom Providers

CUSTOM PROVIDERS CAN BE DEVELOPED TO EXTEND TERRAFORM'S FUNCTIONALITY TO WORK WITH ADDITIONAL APIS.

```
provider "custom"
{ version = "0.1"
endpoint = "https://api.custom.com" }
```

A CUSTOM PROVIDER DEFINITION FOR INTERACTING WITH A CUSTOM API.

Shwetank Singh
GritSetGrow - GSGLearn.com

Dependency Lock File

THE DEPENDENCY LOCK FILE (.TERRAFORM.LOCK.HCL) ENSURES THE CONSISTENT RESOLUTION OF PROVIDER VERSIONS.

N/A

TERRAFORM AUTOMATICALLY GENERATES THE LOCK FILE TO MAINTAIN VERSION CONSISTENCY ACROSS ENVIRONMENTS.

Shwetank Singh
GritSetGrow - GSGLearn.com

Remote Execution

REMOTE EXECUTION ALLOWS YOU TO EXECUTE TERRAFORM RUNS IN A REMOTE ENVIRONMENT, SUCH AS TERRAFORM CLOUD.

```
terraform
  {
   cloud
     { organization = "my-org"
        workspaces
           {
              name = "my-workspace" } }
}
```

CONFIGURES TERRAFORM TO RUN REMOTELY IN TERRAFORM CLOUD.

Shwetank Singh
GritSetGrow - GSGLearn.com

Versioning

VERSIONING ALLOWS YOU TO SPECIFY WHICH VERSIONS OF TERRAFORM OR A PROVIDER YOU ARE USING, ENSURING COMPATIBILITY.

```
terraform { required_version = ">= 0.14" }
```

ENSURES TERRAFORM VERSION 0.14 OR HIGHER IS USED.

Shwetank Singh
GritSetGrow - GSGLearn.com

Error Handling

TERRAFORM PROVIDES MECHANISMS LIKE TRY AND CAN TO HANDLE ERRORS AND UNEXPECTED BEHAVIOR IN EXPRESSIONS.

```
locals { result = try(1 / 0,
"default_value") }
```

USES TRY TO CATCH AN ERROR AND RETURN A DEFAULT VALUE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Sentinel Policies

SENTINEL IS A POLICY-AS-CODE FRAMEWORK THAT CAN ENFORCE COMPLIANCE AND SECURITY POLICIES IN TERRAFORM.

N/A

SENTINEL POLICIES ALLOW YOU TO ENFORCE GUARDRAILS ON TERRAFORM RUNS, ENSURING SECURITY OR COMPLIANCE STANDARDS ARE MET.

Shwetank Singh
GritSetGrow - GSGLearn.com

Data Engineering 101: Terraform Basics

Import Existing Infrastructure

TERRAFORM CAN IMPORT EXISTING RESOURCES, ALLOWING YOU TO MANAGE THEM WITHOUT REBUILDING.

```
terraform import
aws_vpc.example vpc-123456
```

IMPORTS AN EXISTING VPC INTO TERRAFORM'S STATE FILE FOR MANAGEMENT.

Shwetank Singh
GritSetGrow - GSGLearn.com

Interpolation

INTERPOLATION ALLOWS YOU TO REFERENCE VARIABLES AND RESOURCES DYNAMICALLY IN TERRAFORM CONFIGURATION.

```
resource "aws_instance" "example"
{ ami = var.ami_id
instance_type = var.instance_type }
```

DYNAMICALLY INJECTS VARIABLE VALUES INTO THE EC2 INSTANCE RESOURCE DEFINITION.

Shwetank Singh
GritSetGrow - GSGLearn.com

Resource Targeting

YOU CAN TARGET SPECIFIC RESOURCES DURING AN APPLY OR DESTROY OPERATION TO LIMIT THE CHANGES TO ONLY THOSE RESOURCES.

```
terraform apply -
target=aws_instance.example
```

APPLIES CHANGES ONLY TO THE TARGETED EC2 INSTANCE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Outputs with Sensitive Data

OUTPUTS MARKED AS SENSITIVE PREVENT THE VALUE FROM BEING DISPLAYED IN THE CONSOLE, ENSURING DATA LIKE PASSWORDS AREN'T EXPOSED.

```
output "db_password"
{
value = aws_rds_instance.db.password
sensitive = true }
```

ENSURES THAT THE DATABASE PASSWORD ISN’T PRINTED TO THE CONSOLE WHEN APPLYING CHANGES.

Shwetank Singh
GritSetGrow - GSGLearn.com

Built-in Functions

TERRAFORM PROVIDES BUILT-IN FUNCTIONS LIKE FILE(), LENGTH(), AND CONCAT() TO PROCESS AND MANIPULATE DATA WITHIN THE CONFIGURATION.

```
locals { file_content = file("config.json") }
```

READS THE CONTENT OF A FILE INTO A LOCAL VARIABLE.

Shwetank Singh
GritSetGrow - GSGLearn.com

Looping over Maps and Lists

TERRAFORM ALLOWS YOU TO LOOP OVER MAPS OR LISTS USING FOR EXPRESSIONS TO DYNAMICALLY CREATE MULTIPLE RESOURCES OR VALUES.

```
locals {
instance_map = { "app" = "t2.micro",
"db" = "t2.large" } }
```

LOOPS OVER A MAP TO DYNAMICALLY ASSIGN INSTANCE TYPES FOR DIFFERENT APPLICATIONS.

Shwetank Singh
GritSetGrow - GSGLearn.com

Default Values

VARIABLES CAN HAVE DEFAULT VALUES, WHICH ARE USED IF NO OTHER VALUE IS PROVIDED.

```
variable "instance_type" {
default = "t2.micro" }
```

SETS A DEFAULT INSTANCE TYPE OF T2.MICRO IF NO VALUE IS PROVIDED DURING THE TERRAFORM RUN.

Shwetank Singh
GritSetGrow - GSGLearn.com

THANK you

Like

Share

Follow