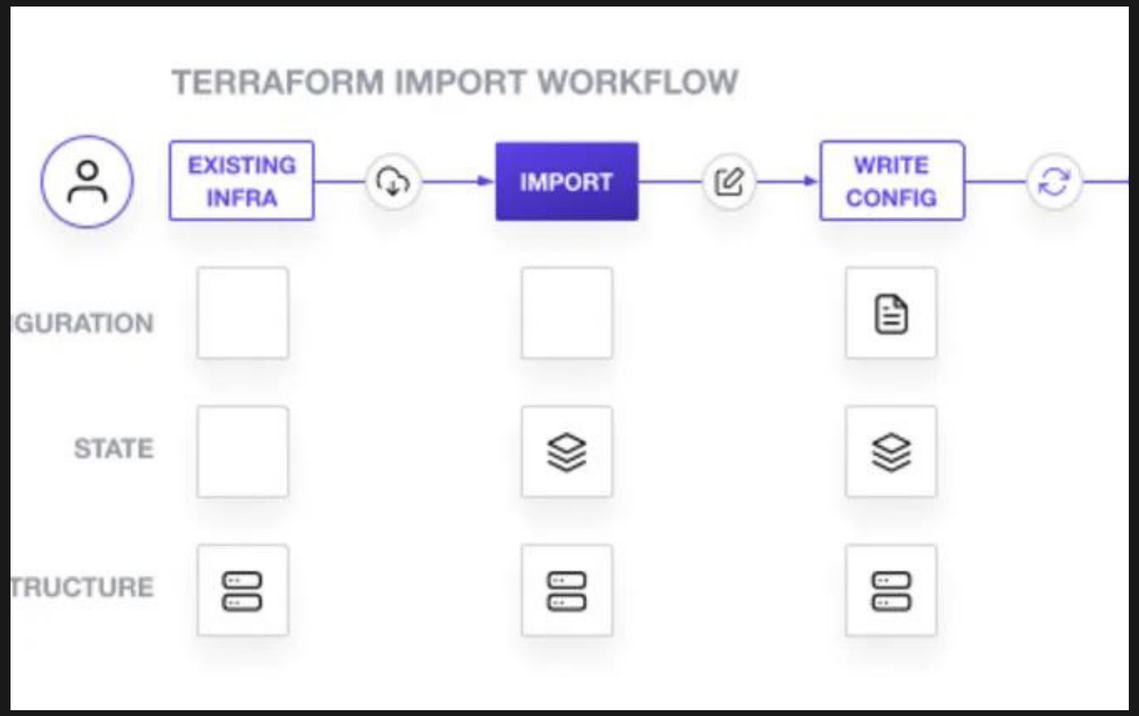
Terraform State Import

HashiCorp Terraform

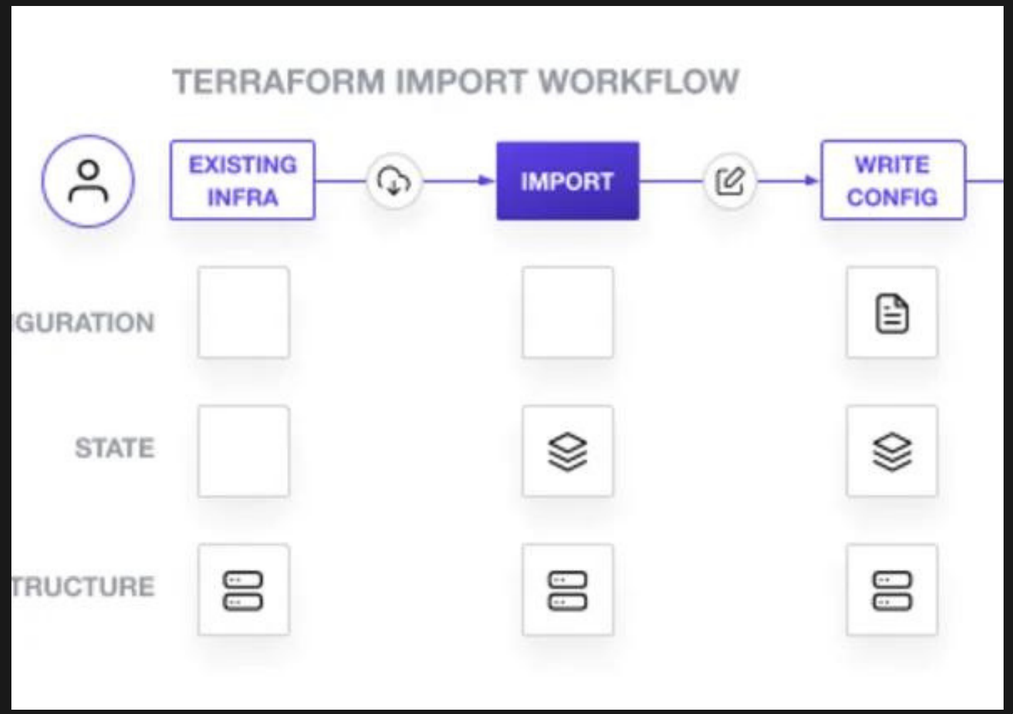
State Import



HashiCorp Terraform

State Import

- Terraform uses state to manage all provisioned resources
  - But what if resource was created manually, and needs to be managed by Terraform?
- Importing resource into state file allows Terraform to manage and track resource which was created manually
- This is common in refactoring
  a. Infrastructure gets created manually for quick testing
  b. Terraform config for infrastructure type gets created
  c. Infrastructure now needs to be imported so it's managed by Terraform & achieves IaC best practice



HashiCorp Terraform

State Import

- Before importing resource into Terraform state, valid resource config must already exist
  - If importing a VPC subnet, a resource definition for that VPC subnet must already be defined
  - Like a parasite which needs a host
  - If we run terraform apply before import, then Terraform will propose to create a new subnet
- In example, we have resource definition for subnet: aws_subnet.main (remember resource addressing)
  - We want to import a subnet created manually into this resource definition

```
resource "aws_subnet" "main" {
  vpc_id     = aws_vpc.main.id
  cidr_block = "10.0.1.0/24"

  tags = {
    Name = "Main"
  }
}
```

HashiCorp Terraform

State Import

- terraform import <resource_address> <address ID>
  - Supply the command with the resource address of the defined resource (the host)
  - Provide the address ID, which is the unique identifier in AWS for that specific resource
  - The address ID depends on the type of resource being imported
  - For AWS VPC Subnets, the address ID is the subnet id
  - terraform import aws_subnet.public_subnet subnet-9d4a7b6c
  - Refer to terraform docs on how to import that resource into Terraform
- After an import, Terraform will create a new entry into state file, using that resource address