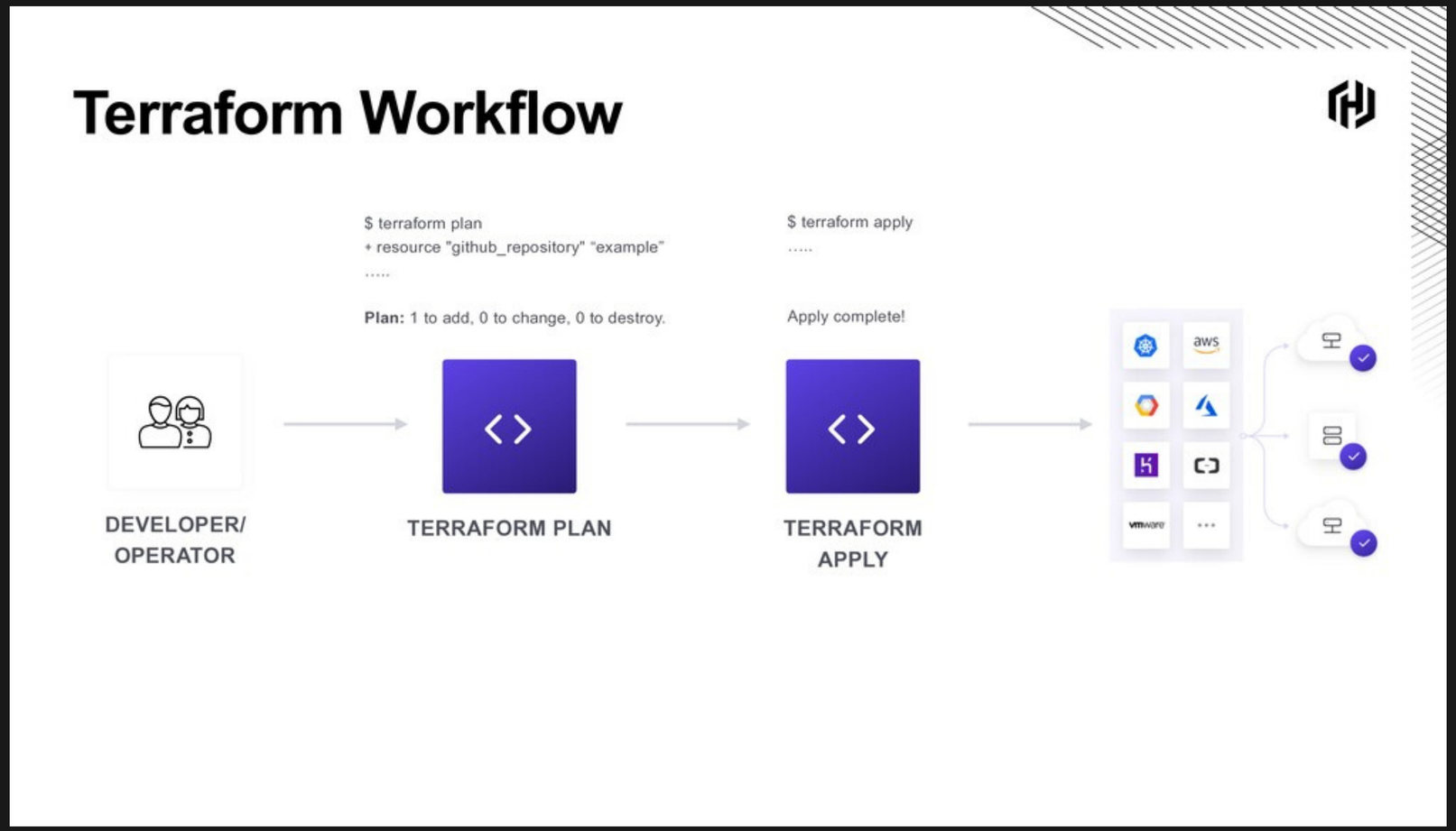
Terraform Workflow Introduction

HashiCorp Terraform

Terraform Workflow



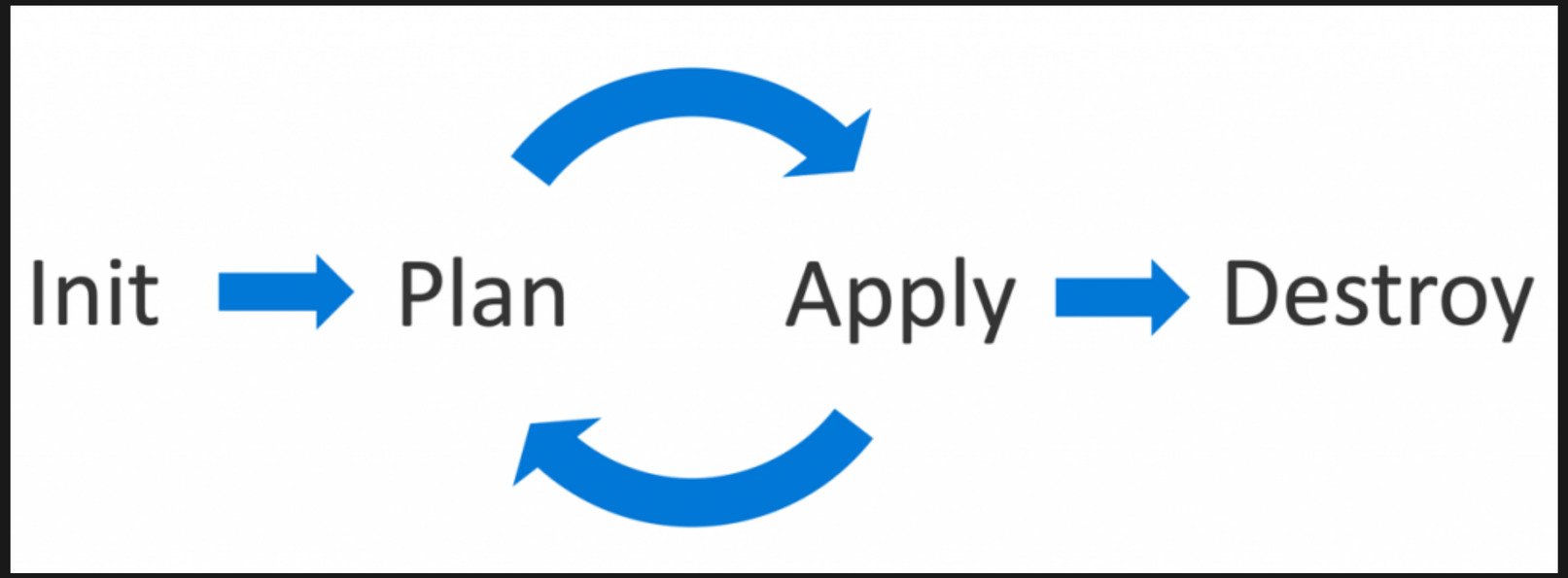
HashiCorp
Terraform

Terraform Workflow

Workflow Fundamentals

- There are two ways of using Terraform:
  a. Terraform CLI - The Terraform CLI commands used to view, apply or destroy resources based on what's defined in code, or to view and manipulate the state. We call this the Core Workflow.
  b. Writing Terraform - Writing actual Terraform code to define new infrastructure resources
- In the real world, you write Terraform code, then apply your changes with the Terraform CLI
- This section covers the first method, using the Terraform CLI - going over the most important commands to interact with Terraform
- The Terraform CLI provides you with important commands, to manage your infrastructure
- The general workflow goes like this (usually in order):
  c. terraform init - Initializes a directory, pulling in recent provider and module changes
  d. terraform plan - terraform reads your code, to perform a dry-run (does not make any changes)
  e. terraform apply - terraform reads your current code, and applies the necessary changes
  f. terraform destroy - terraform reads your current code, to destroy infrastructure

Terraform Workflow



HashiCorp Terraform

Terraform Workflow

```
resource "aws_instance" "main" {
  provider = alias.west
  ami = data.aws_ami.aws_linux_2.id

  instance_type               = "t3.small"
  associate_public_ip_address = true

  vpc_security_group_ids = [aws_security_group.allow_ssh.id]

  ebs_block_device {
    device_name = "/dev/sdf"
    volume_type = "gp2"
    volume_size = "2"
  }

  tags = merge(local.tags, {
    Name : "levelup_with_terraform"
  })
}
```

- The general workflow, to apply this Terraform code:
  a. Run terraform init - keeps project updated with latest configuration, provider, and module changes. Initializes the defined state backend.
  b. Run terraform plan - performs a dry run, to view the effects of your code before applying them
  c. Run terraform apply - after validating the plan output, and if it all looks good, apply your changes by creating this resource
  d. Run terraform destroy - to perform cleanup, or if the resource is no longer needed

HashiCorp Terraform

Terraform Workflow
Init Demo

Terraform Workflow
Plan Demo

Terraform Workflow
Apply Demo

Terraform Workflow
Destroy Demo