

Manual drift is a story nobody can review.

Infrastructure as Code makes the desired end state visible before anyone changes the live environment. Terraform gives that change a plan, a checkpoint, and an execution path.

01
DEFINE INTENT

02
REVIEW PLAN

03
APPLY WITH CARE

TAMRA'S READ

A plan is not paperwork. It is the moment a change becomes discussable.

The difference between a manual click and an operational decision is often a reviewable record.

01 / THE CONTRAST

From manual drift to controlled change

IaC is a team practice: define the target, preview the impact, approve the decision, then observe the result.

THE MANUAL DRIFT LOOP

1. A change happens in a console
2. Documentation falls behind
3. The next change starts from guesswork
4. Drift becomes a surprise

THE IaC CONTROL LOOP

1. Configuration states the intent
2. Validation checks assumptions
3. Plan shows proposed impact
4. Approval gates the apply

What Terraform contributes

Terraform reads configuration and state, calculates a proposed change, then uses providers to communicate with target platform APIs. The plan is the handoff between intent and action.

WRITE	PLAN	APPROVE	APPLY	OBSERVE
.tf configuration	reviewable diff	decision point	provider APIs	real environment

THE REVIEW QUESTIONS

- **Does this touch only the intended environment?**
- **Are replacements, deletions, access changes, and costs understood?**
- **Who owns recovery if the plan does not land as expected?**

Practice the rhythm before production

Start locally. The example below uses Terraform's terraform_data resource, so it does not need a cloud provider or billable resource.

main.tf

```
resource "terraform_data" "hello" {
  input = "Hello, IaC"
}

# no cloud provider required
```

```
$ terraform init
$ terraform fmt && terraform validate
$ terraform plan
```

PLAN REVIEW

Scope
Intended environment only

Impact
Deletes or replacements understood

Access
Roles and data paths checked

Recovery
Rollback path documented

Production guardrails

STATE
Use a protected shared backend. Treat state as sensitive operational data.

SECRETS
Keep credentials and private values out of code and source control.

GUARDRAILS
Use policy checks and a peer review before high-impact apply.

SMALL START
Begin in a bounded non-production environment.

OBSERVE
Verify the actual platform after apply.

THE BOTTOM LINE

IaC is controlled change, not faster clicking.

Terraform can make infrastructure work repeatable across teams and services. It cannot replace architecture decisions, accountable ownership, or careful review.