Resources and Data Sources

Resources are the building blocks

Every piece of Azure infrastructure you want Terraform to manage is defined as a **resource block**.

Defining a resource

Here's a resource block that creates an Azure resource group:

```
resource "azurerm_resource_group" "rg" {
  name     = "rg-terraform-demo"
  location = "uksouth"
}
```

Three parts to every resource block:

- **Resource type** — `"azurerm_resource_group"` (provider + resource)
- **Local name** — `"rg"` (used to reference this resource elsewhere)
- **Configuration block** — the properties for that resource

Referencing one resource from another

Resources can reference each other's properties directly:

```
resource "azurerm_resource_group" "rg" {
  name     = "rg-terraform-demo"
  location = "uksouth"
}

resource "azurerm_storage_account" "sa" {
  name                     = "sttfdemo${random_string.suffix.result}"
  resource_group_name      = azurerm_resource_group.rg.name
  location                 = azurerm_resource_group.rg.location
  account_tier             = "Standard"
  account_replication_type = "LRS"
}
```

`azurerm_resource_group.rg.name` follows the pattern `<type>.<local_name>.<attribute>`.

Why references matter

When the storage account references the resource group, Terraform understands there's a **dependency** between them.

It will always create the resource group **first**, without you having to specify the order.

> *This is one of Terraform's most powerful features — dependency resolution is automatic.*

What is a data source?

Data sources let you **read** information about resources that already exist in Azure — things Terraform didn't create and doesn't manage.

```
data "azurerm_resource_group" "existing" {
  name = "rg-existing"
}
```

Use the `data` keyword instead of `resource`. Terraform queries Azure and makes the properties available to your configuration.

Using a data source

Reference a data source with `data.<type>.<name>.<attribute>`:

```
resource "azurerm_storage_account" "sa" {
  name                     = "sttfdemo${random_string.suffix.result}"
  resource_group_name      = data.azurerm_resource_group.existing.name
  location                 = data.azurerm_resource_group.existing.location
  account_tier             = "Standard"
  account_replication_type = "LRS"
}
```

> *The resource group already exists in Azure — Terraform reads it but never modifies or deletes it.*

When to use a data source vs a resource

Scenario	Use
Terraform should create and manage it	`resource`
It already exists, you just need its properties	`data`
Created by another team's Terraform state	`data`

A common example: referencing an existing Key Vault or ACR that was provisioned separately.

Finding available properties

Every resource and data source has documented attributes. Check the Azure Provider documentation to see what's available.

For example, `azurerm_resource_group` exposes:

- `name`
- `location`
- `id`
- `tags`

The docs also tell you which attributes are **required**, which are **optional**, and which are **read-only** (computed by Azure after creation).