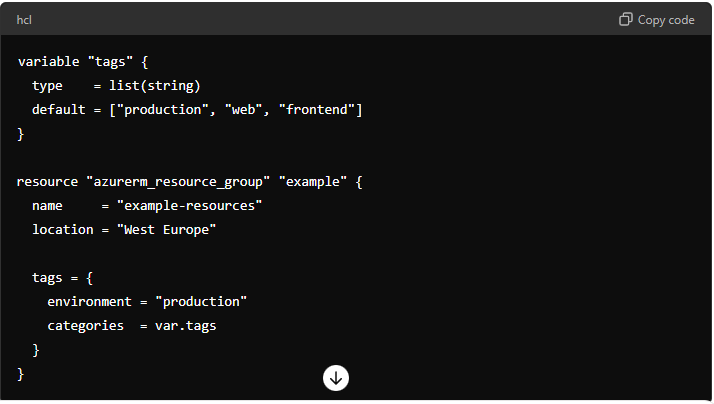
In Terraform, list, set, and map are different types of data structures used to manage collections of values. These are crucial for defining and organizing various resources and configurations in cloud providers like Azure. Let's explore each of them with examples related to Azure resources:

**1. List in Terraform**

A list in Terraform is an ordered collection of elements where each element can be accessed using its index. Lists are defined using square brackets **[ ]** and elements are separated by commas.

**Example using Azure:**

variable "tags" {

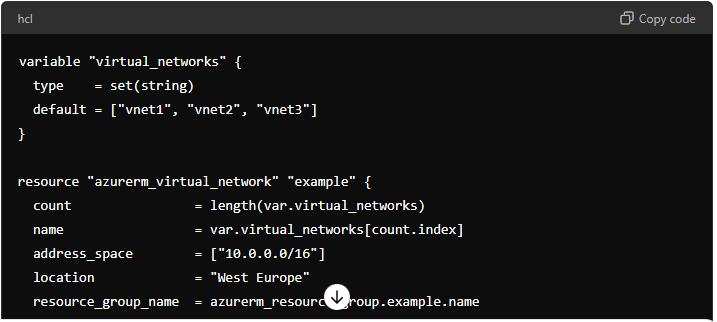


**Explanation:** In this example, tags is a list variable containing strings. The **azurerm\_resource\_group** resource uses this list to set tags on the Azure resource group. Each element in the list is assigned a tag category, such as production, web, and frontend.

### 2. Set in Terraform

A set in Terraform is an unordered collection of unique elements. Sets are defined using braces { } and elements are separated by commas.

**Example using Azure:**

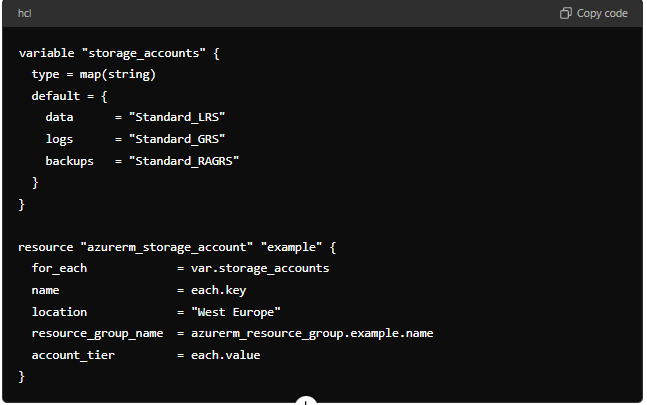


**Explanation:** Here, virtual\_networks is a set variable containing strings representing names of virtual networks. The azurerm\_virtual\_network resource is created multiple times (equal to the number of elements in the set) to deploy each virtual network with a unique name.

### 3. Map in Terraform

A map in Terraform is a collection of key-value pairs where each key is unique. Maps are defined using braces { } with each pair written as key = value.

**Example using Azure:**



**Explanation:** In this example, storage\_accounts is a map variable where keys represent different types of storage (data, logs, backups) and values represent the SKU (Standard\_LRS, Standard\_GRS, Standard\_RAGRS). The azurerm\_storage\_account resource is deployed for each key-value pair in the map, creating multiple storage accounts with different configurations.

### Summary:

* **Lists** are ordered collections used for maintaining sequence or order.
* **Sets** are unordered collections where each element is unique, useful for managing unique items like resource names.
* **Maps** are collections of key-value pairs used for associating related data, such as configuration settings or resource parameters.

These data structures in Terraform provide flexibility and efficiency when managing Azure resources and configurations, allowing you to define and organize your infrastructure in a structured manner.