**About the Terraform Language**

**How to Use Terraform Language**

The main purpose of the Terraform language is to declare resources, which represent infrastructure objects. All other language features exist only to make the definition of resources more flexible and convenient.

A Terraform configuration is a complete document in the Terraform language that tells Terraform how to manage a given collection of infrastructure. A configuration can consist of multiple files and directories.

The syntax of the Terraform language consists of only a few basic elements:

resource "aws\_vpc" "main" {

cidr\_block = var.base\_cidr\_block

}

< BLOCK TYPE > "< BLOCK LABEL >" "< BLOCK LABEL >" {

# Block body

< IDENTIFIER > = < EXPRESSION > # Argument

}

The Terraform language is declarative, describing an intended goal rather than the steps to reach that goal. The ordering of blocks and the files they are organized into are generally not significant; Terraform only considers implicit and explicit relationships between resources when determining an order of operations.

**Resource Syntax**

Resources are the most important element in the Terraform language. Each resource block describes one or more infrastructure objects, such as virtual networks, compute instances, or higher-level components such as DNS records.

Resource declarations can include a number of advanced features, but only a small subset is required for initial use. More advanced syntax features, such as single resource declarations that produce multiple similar remote objects, are described later on this page.

#example.tf

resource "aws\_instance" "web" {

ami = "ami-a1b2c3d4"

instance\_type = "t2.micro"

}

A resource block declares a resource of a given ***type*** ("aws\_instance") with a given ***local name*** ("web"). The name is used to refer to this resource from elsewhere in the same Terraform module, but has no significance outside that module's scope.

The resource type and name together serve as an identifier for a given resource and so must be unique within a module.

Within the block body (between { and }) are the ***configuration arguments*** for the resource itself. Most arguments in this section depend on the resource type, and indeed in this example both ami and instance\_type are arguments defined specifically for the aws\_instance resource type.

Note: Resource names must start with a letter or underscore, and may contain only letters, digits, underscores, and dashes.

Every Terraform provider has its own documentation, describing its resource types and their arguments.

Most publicly available providers are distributed on the ***Terraform Registry***, which also hosts their documentation. When viewing a provider's page on the Terraform Registry, you can click the "**Documentation**" link in the header to browse its documentation. Provider documentation on the registry is versioned, and you can use the dropdown version menu in the header to switch which version's documentation you are viewing.

Formun Üstü

Formun Altı