Terraform Best Practices

**TABLE OF CONTENTS**

[Best Practices 1](#_Toc1720439405)

[Naming 1](#_Toc563187559)

[General conventions 1](#_Toc1140858885)

[Service Users 1](#_Toc218798029)

[Resource and data source arguments 1](#_Toc896516024)

[Variables 2](#_Toc8454459)

[Outputs 2](#_Toc852868051)

[Resources: 2](#_Toc9478135)

# Best Practices

## Naming

From <https://www.terraform-best-practices.com/naming>

## General conventions

There should be no reason to not follow at least these 😃

Use \_ not -

* **MUST** Use \_ (underscore) instead of - (dash) in all: resource names, data source names, variable names, outputs.
* **EXCEPTION** Beware that actual cloud resources have many hidden restrictions in their naming conventions. Some cannot contain dashes, some must be camel cased. These conventions refer to Terraform names themselves. (aka for s3 buckets use camelCase)
* **MUST** Only use lowercase letters and numbers.
* repo structure **MUST** follow Standard Structure  
  <https://www.terraform.io/language/modules/develop/structure>
* $ tree complete-module/.├── README.md├── main.tf├── variables.tf├── outputs.tf├── ...├── modules/│ ├── nestedA/│ │ ├── README.md│ │ ├── variables.tf│ │ ├── main.tf│ │ ├── outputs.tf│ ├── nestedB/│ ├── .../├── examples/│ ├── exampleA/│ │ ├── main.tf│ ├── exampleB/│ ├── .../

## Service Users

* ALL service users MUST be post fix "\_service" to help operations determine purpose of account

## Resource and data source arguments

Do not repeat resource type in resource name (not partially, nor completely):

* Good: resource "aws\_route\_table" "public" {}
* Bad: resource "aws\_route\_table" "public\_route\_table" {}
* Bad: resource "aws\_route\_table" "public\_aws\_route\_table" {}
* Resource name should be named this if there is no more descriptive and general name available, or if resource module creates single resource of this type (eg, there is single resource of type aws\_nat\_gateway and multiple resources of typeaws\_route\_table, so aws\_nat\_gateway should be named this and aws\_route\_table should have more descriptive names - like private, public, database).
* Always use singular nouns for names.
* Use - inside arguments values and in places where value will be exposed to a human (eg, inside DNS name of RDS instance).
* Include count argument inside resource blocks as the first argument at the top and separate by newline after it. See example.
* Include tags argument, if supported by resource as the last real argument, following by depends\_on and lifecycle, if necessary. All of these should be separated by single empty line. See example.
* When using condition in count argument use boolean value, if it makes sense, otherwise use length or other interpolation. See example.
* To make inverted conditions don't introduce another variable unless really necessary, use 1 - boolean value instead. For example, count = "${1 - var.create\_public\_subnets}"

## Variables

* Don't reinvent the wheel in resource modules - use the same variable names, description and default as defined in "Argument Reference" section for the resource you are working on.
* Omit type = "list" declaration if there is default = [] also.
* Omit type = "map" declaration if there is default = {} also.
* Use plural form in name of variables of type list and map.
* When defining variables order the keys: description , type, default .
* Always include description for all variables even if you think it is obvious.

## Outputs

Name for the outputs is important to make them consistent and understandable outside of its scope (when user is using a module it should be obvious what type and attribute of the value is returned).

* The general recommendation for the names of outputs is that it should be descriptive for the value it contains and be less free-form than you would normally want.
* Good structure for names of output looks like {name}*{type}*{attribute} , where:  
  {name} is a resource or data source name without provider prefix. {name} for aws\_subnet is subnet, foraws\_vpc it is vpc.  
  {type} is a type of a resource sources  
  {attribute} is an attribute returned by the output  
  See examples.
* If output is returning a value with interpolation functions and multiple resources, the {name} and {type} there should be as generic as possible (this is often the most generic and should be preferred). See example.
* If the returned value is a list it should have plural name. See example.
* Always include description for all outputs even if you think it is obvious.

# Resources:

<https://www.terraform.io/docs/cloud/guides/recommended-practices/index.html>

<https://www.terraform-best-practices.com/naming>