

Note that even though you should almost always prefer `for_each` over `count` for creating multiple copies of a resource or module, when it comes to conditional logic, setting `count` to 0 or 1 tends to be simpler than setting `for_each` to an empty or nonempty collection. Therefore, I typically recommend using `count` to conditionally create resources and modules, and using `for_each` for all other types of loops and conditionals.

Conditionals with the `if` String Directive

Let's now look at the `if` string directive, which has the following syntax:

```
%{ if <CONDITION> }<TRUEVAL>%{ endif }
```

where `CONDITION` is any expression that evaluates to a boolean and `TRUEVAL` is the expression to render if `CONDITION` evaluates to true.

Earlier in the chapter, you used the `for` string directive to do loops within a string to output several comma-separated names. The problem was that there was an extra trailing comma and space at the end of the string. You can use the `if` string directive to fix this issue as follows:

```
output "for_directive_index_if" {
    value = <<EOF
%{ for i, name in var.names }
    ${name} %{ if i < length(var.names) - 1 }, %{ endif }
%{ endfor }
EOF
}
```

There are a few changes here from the original version:

- I put the code in a *HEREDOC*, which is a way to define multiline strings. This allows me to spread the code out across several lines so it is more readable.
- I used the `if` string directive to not output the comma and space for the last item in the list.

When you run `terraform apply`, you get the following output:

```
$ terraform apply  
(...)  
  
Outputs:  
  
for_directive_index_if = <<EOT  
  
    neo,  
  
    trinity,  
  
    morpheus  
  
EOT
```

Whoops. The trailing comma is gone, but we've introduced a bunch of extra whitespace (spaces and newlines). Every whitespace you put in a HEREDOC ends up in the final string. You can fix this by adding *strip markers* (~) to your string directives, which will eat up the extra whitespace before or after the strip marker:

```
output "for_directive_index_if_strip" {  
  value = <<EOF  
~{~ for i, name in var.names ~}  
${name}{ if i < length(var.names) - 1 }, ~{ endif }  
~{~ endfor ~}  
EOF  
}
```

Let's give this version a try:

```
$ terraform apply  
(...)  
  
Outputs:  
  
for_directive_index_if_strip = "neo, trinity, morpheus"
```