

Let's go through these, one at a time.

## Conditionals with the count Parameter

The `count` parameter you saw earlier lets you do a basic loop. If you're clever, you can use the same mechanism to do a basic conditional. Let's begin by looking at if-statements in the next section and then move on to if-else-statements in the section thereafter.

### If-statements with the count parameter

In [Chapter 4](#), you created a Terraform module that could be used as a “blueprint” for deploying web server clusters. The module created an Auto Scaling Group (ASG), Application Load Balancer (ALB), security groups, and a number of other resources. One thing the module did *not* create was the scheduled action. Because you want to scale the cluster out only in production, you defined the `aws_autoscaling_schedule` resources directly in the production configurations under *live/prod/services/webserver-cluster/main.tf*. Is there a way you could define the `aws_autoscaling_schedule` resources in the `webserver-cluster` module and conditionally create them for some users of the module and not create them for others?

Let's give it a shot. The first step is to add a Boolean input variable in *modules/services/webserver-cluster/variables.tf* that you can use to specify whether the module should enable auto scaling:

```
variable "enable_autoscaling" {  
  description = "If set to true, enable auto scaling"  
  type        = bool  
}
```

Now, if you had a general-purpose programming language, you could use this input variable in an if-statement:

```
# This is just pseudo code. It won't actually work in Terraform.  
if var.enable_autoscaling {  
  resource "aws_autoscaling_schedule"
```