Module Outputs

A powerful feature of ASGs is that you can configure them to increase or decrease the number of servers you have running in response to load. One way to do this is to use a *scheduled action*, which can change the size of the cluster at a scheduled time during the day. For example, if traffic to your cluster is much higher during normal business hours, you can use a scheduled action to increase the number of servers at 9 a.m. and decrease it at 5 p.m.

If you define the scheduled action in the `webserver-cluster` module, it would apply to both staging and production. Because you don't need to do this sort of scaling in your staging environment, for the time being, you can define the auto scaling schedule directly in the production configurations (in Chapter 5, you'll see how to conditionally define resources, which lets you move the scheduled action into the `webserver-cluster` module).

To define a scheduled action, add the following two `aws_autoscaling_schedule` resources to *prod/services/webserver-cluster/main.tf*:

```
resource "aws_autoscaling_schedule"
"scale_out_during_business_hours" {
  scheduled_action_name = "scale-out-during-business-hours"
  min_size              = 2
  max_size              = 10
  desired_capacity      = 10
  recurrence            = "0 9 * * *"
}

resource "aws_autoscaling_schedule" "scale_in_at_night" {
  scheduled_action_name = "scale-in-at-night"
  min_size              = 2
  max_size              = 10
  desired_capacity      = 2
  recurrence            = "0 17 * * *"
}
```

This code uses one `aws_autoscaling_schedule` resource to increase the number of servers to 10 during the morning hours (the `recurrence` parameter uses cron syntax, so `"0 9 * * *"` means "9 a.m. every day") and a second `aws_autoscaling_schedule` resource to decrease the number of servers at night (`"0 17 * * *"` means "5 p.m. every day"). However, both usages of `aws_autoscaling_schedule` are missing a required parameter, `autoscaling_group_name`, which specifies the name of the ASG. The ASG itself is defined within the `webserver-cluster` module, so how do you access its name? In a general-purpose programming language such as Ruby, functions can return values:

```
# A function that returns a value
def example_function(param1, param2)
  return "Hello, #{param1} #{param2}"
end

# Call the function and get the return value
return_value = example_function("foo", "bar")
```

In Terraform, a module can also return values. Again, you do this using a mechanism you already know: output variables. You can add the ASG name as an output variable in */modules/services/webserver-cluster/outputs.tf* as follows:

```
output "asg_name" {
  value       = aws_autoscaling_group.example.name
  description = "The name of the Auto Scaling Group"
}
```

You can access module output variables using the following syntax:

```
module.<MODULE_NAME>.<OUTPUT_NAME>
```

For example:

```
module.frontend.asg_name
```

In *prod/services/webserver-cluster/main.tf*, you can use this syntax to set the `autoscaling_group_name` parameter in each of the `aws_autoscaling_schedule` resources:

```
resource "aws_autoscaling_schedule"
"scale_out_during_business_hours" {
  scheduled_action_name = "scale-out-during-business-hours"
  min_size              = 2
  max_size              = 10
  desired_capacity      = 10
  recurrence            = "0 9 * * *"

  autoscaling_group_name = module.webserver_cluster.asg_name
}

resource "aws_autoscaling_schedule" "scale_in_at_night" {
  scheduled_action_name = "scale-in-at-night"
  min_size              = 2
  max_size              = 10
  desired_capacity      = 2
  recurrence            = "0 17 * * *"

  autoscaling_group_name = module.webserver_cluster.asg_name
}
```

You might want to expose one other output in the `webserver-cluster` module: the DNS name of the ALB, so you know what URL to test when the cluster is deployed. To do that, you again add an output variable in */modules/services/webserver-cluster/outputs.tf*:

```
output "alb_dns_name" {
  value       = aws_lb.example.dns_name
  description = "The domain name of the load balancer"
}
```

You can then "pass through" this output in *stage/services/webserver-cluster/outputs.tf* and *prod/services/webserver-cluster/outputs.tf* as follows:

```
output "alb_dns_name" {
  value       = module.webserver_cluster.alb_dns_name
  description = "The domain name of the load balancer"
}
```