**Outputs**

**Overview**

In the previous section, we introduced input variables as a way to parameterize Terraform configurations. In this page, we introduce output variables as a way **to organize data to be easily queried and shown back to the Terraform user**.

When building potentially complex infrastructure, Terraform stores hundreds or thousands of attribute values for all your resources. But as a user of Terraform, you may only be interested in a few values of importance, such as *a load balancer IP, VPN address, etc.*

Outputs are a way to tell Terraform *what data is important*. This data is outputted when apply is called, and can be queried using the terraform output command.

**Defining Outputs**

Let's define an output to show us the public IP address of the elastic IP address that we create. Add this to any of your \*.tf files:

output "ip" {

value = aws\_eip.ip.public\_ip

}

This defines an output variable named "ip". The value field specifies what the value will be, and almost always contains one or more interpolations, since the output data is typically dynamic. In this case, we're outputting the public\_ip attribute of the elastic IP address.

Multiple output blocks can be defined to specify multiple output variables.

### Viewing Outputs

Run terraform apply to populate the output. This only needs to be done once after the output is defined. The apply output should change slightly. At the end you should see this:

$ terraform apply

...

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

ip = 50.17.232.209

apply highlights the outputs. You can also query the outputs after apply-time using terraform output:

$ terraform output ip

50.17.232.209

This command is useful for scripts to extract outputs.