

Hands-on Infrastructure Automation with Terraform on AWS

Alexander Savchuk

Building a multi-tier environment

Section 4

Organising data with output variables

Resources and data sources

- Resources provide **create**, **read**, **update**, and **delete** functionality (CRUD)
- Data sources support **read** operations only

```
data "aws_ami" "amazon_linux" {  
    most_recent = true  
  
    filter {  
        name      = "name"  
        values    = [ "amzn-ami-*-x86_64-gp2" ]  
    }  
}
```

Resources and data sources

- Resources provide **create**, **read**, **update**, and **delete** functionality (CRUD)
- Data sources support **read** operations only

```
data "aws_ami" "amazon_linux" {  
    most_recent = true  
  
    filter {  
        name      = "name"  
        values    = [ "amzn-ami-*-x86_64-gp2" ]  
    }  
}
```

Output variables

- Expose important resource attributes and make them easier to query
- Outputs are exposed to the user and stored in the state file during terraform apply.
- A single output block configures a single variable.

Example output

```
output "vpc_id" {  
    value = "${aws_vpc.main.id}"  
    description = "VPC id"  
    sensitive = false  
}
```

Example output

```
output "vpc_id" {  
    value = "${aws_vpc.main.id}"  
    description = "VPC id"  
    sensitive = false  
}
```


Example output

```
output "vpc_id" {  
    value = "${aws_vpc.main.id}"  
    description = "VPC id"  
    sensitive = false  
}
```

Example output

```
output "vpc_id" {  
    value = "${aws_vpc.main.id}"  
    description = "VPC id"  
    sensitive = false  
}
```

Integrating components in a
complex environment