Terraform Data Sources

Data Sources

- Data sources are a way of retrieving resource data from outside of Terraform
  - Data created outside of your current Terraform config
  - For example, when creating an EC2 instance with AWS Linux provided AMI
  - This is done by providing set of filters to tell Terraform which resource to grab
- Data sources also useful for retrieving resources created from other Terraform configurations
  - If your networking resources are created in separate repository, in a separate state file

```
data "aws_ami" "aws_linux_2" {
  most_recent = true

  filter {
    name   = "name"
    values = ["amzn2-ami-hvm*"]
  }

  owners = ["amazon"]
}
```

HashiCorp
Terraform

Data Sources

- Data sources created with using the data block, and providing two arguments
  - The resource type, and the name provided to the data block
  - Similar to how you create a resource
- Most, if not all resource types have a corresponding data type
  - Always refer to documentation to see data type equivalent of a resource

```
data "aws_ami" "example" {
  most_recent = true

  owners = ["self"]
  tags = {
    Name    = "app-server"
    Tested = "true"
  }
}
```

HashiCorp
Terraform

Data Sources

- A powerful feature of Data sources - rendering IAM policies
  - One of the most common use cases for data sources
- Data sources can create IAM policy documents and return a json file which can be read by the aws_iam_policy resource to create an IAM policy

```
data "aws_iam_policy_document" "example" {
  statement {
    sid = "1"

    actions = [
      "s3:ListAllMyBuckets",
      "s3:GetBucketLocation",
    ]

    resources = [
      "arn:aws:s3:::*",
    ]
  }

  statement {
    actions = [
      "s3:ListBucket",
    ]

    resources = [
      "arn:aws:s3:::${var.s3_bucket_name}",
    ]

    condition {
      test     = "StringLike"
      variable = "s3:prefix"

      values = [
        "",
```

```
resource "aws_iam_policy" "example" {
  name   = "example_policy"
  path   = "/"
  policy = data.aws_iam_policy_document.example.json
}
```