

# Data Sources

OW

# Data Sources

- Data sources allow Terraform to use information defined outside of Terraform, defined by another separate Terraform configuration, or modified by functions.
- A data source is accessed via a special kind of resource known as a data resource, declared using a data block:

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

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

• •

- In Terraform, you cannot save the results of a data source block directly.
- Data sources are used to retrieve information dynamically during the Terraform execution, and their values are not saved in the state file or persisted between runs.
- you can capture and save the output of a data source block by using an output block in your Terraform configuration.

# Data block

```
data "aws_instance" "myinstance" {
  filter {
    name     = "tag:Name"
    values = ["test"]
  }
  depends_on = [
    aws_instance.example
  ]
}
output "myoutput" {
  value = data.aws_instance.myinstance.public_ip
}
```

aws\_instance.myinstance

# Query on non-terraform provisioned vm

```
root@jenkins:~/aws# vi output.tf
data "aws_instance" "myinstance" {
  filter {
    name     = "tag:Name"
    values = ["test"]
  }
}
output "myoutput" {
  value = data.aws_instance.myinstance.public_ip
}
~
```

# List the az available in zone

```
provider "aws" {
  access_key = "AKIA43KKKGRBTHIR2T67"
  secret_key = "f/XnR+tAAtuhCLLfeOW+V6Qgla0cc3ZzTK0vRyhS"
  region     = "us-west-2"    # Replace with your desired AWS region
}
/*
resource "aws_instance" "example" {
  ami           = "ami-0b8987a72eee28c3d"    # Replace with your desired AMI ID
  instance_type = "t2.micro"
  subnet_id    = "subnet-0e09953db95a5ac65"    # Replace with your desired subnet ID
  key_name     = "my-key-pair"                # Replace with your desired key pair name
  tags = {
    Name = "instance1"
  }
}
*/
data "aws_availability_zones" "available" {
  state = "available"

  filter {
    name   = "zone-type"
    values = ["availability-zone"]
  }
}

output "available_zones" {
  value = data.aws_availability_zones.available.names
}
```

..

The `aws_availability_zones` data source retrieves information about the availability zones available in your AWS region.

The `state` parameter is set to `"available"` to filter for only the available availability zones. This ensures that only the active zones are considered.

The `filter` block is used to further filter the results. In this case, it filters by the `"zone-type"` attribute and selects availability zones.

# Try

- Create a datasource which list the ami
- owners = ["amazon"]
- Use filter

```
name = "architecture"
```

```
values = ["x86_64"]
```

```
name = "state"
```

```
values = ["available"]
```

```
name = "root-device-type"
```

```
values = ["ebs"]
```