# Adding a New Provider to Your Configuration



Ned Bellavance
HashiCorp Ambassador

@ned1313 nedinthecloud.com

#### Overview



**Globomantics requests** 

**Understanding providers** 

**Dependency graphs** 

Post deployment configuration



#### Globomantics Scenario

#### Potential Improvements



Copy website content

Log traffic to an S3 bucket

Use specific provider versions

**Properly format files** 

#### Potential Improvements



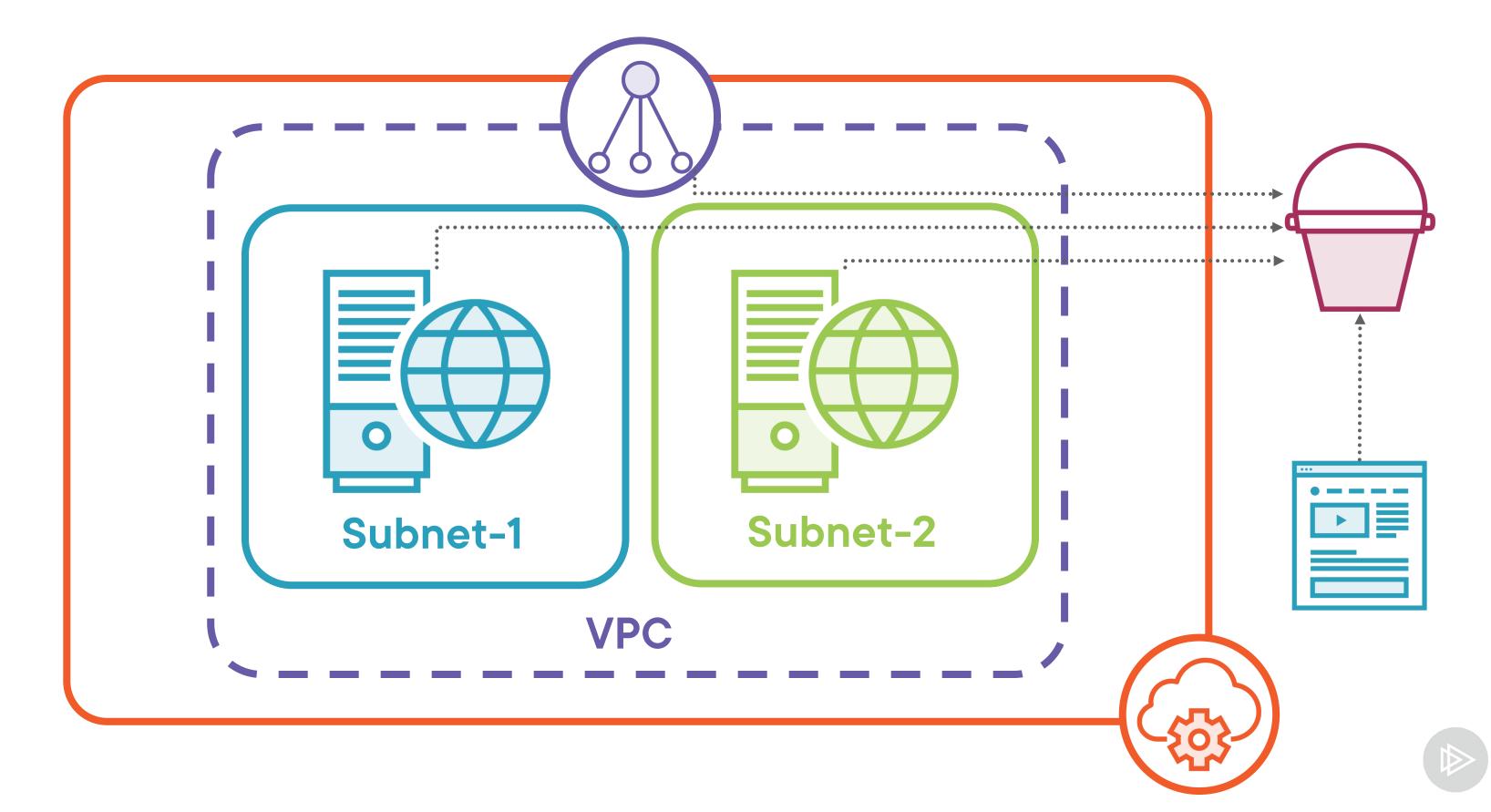
Copy website content

Log traffic to an S3 bucket

Use specific provider versions

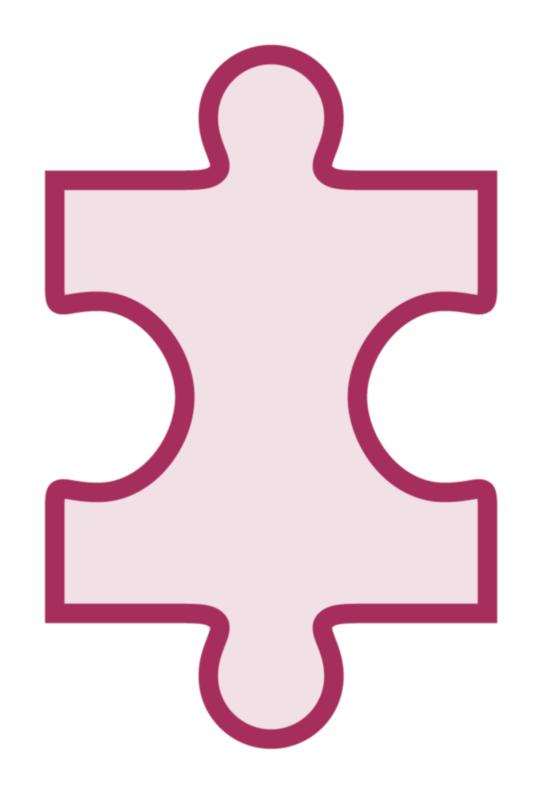
**Properly format files** 

# Deployment Architecture



### Terraform Providers

#### Terraform Providers



Public and private registries

Official, Verified, and Community

Open source

Resources and data sources

Versioned

Multiple instances



#### Terraform Block Syntax

```
terraform {
 required_providers {
  provider_name = {
   source = "address_to_provider"
   # =, <, >, and ~>
   version = "version_expression"
```

# Terraform Block Syntax

```
terraform {
 required_providers {
  aws = {
   source = "hashicorp/aws"
   version = "~>3.0"
```

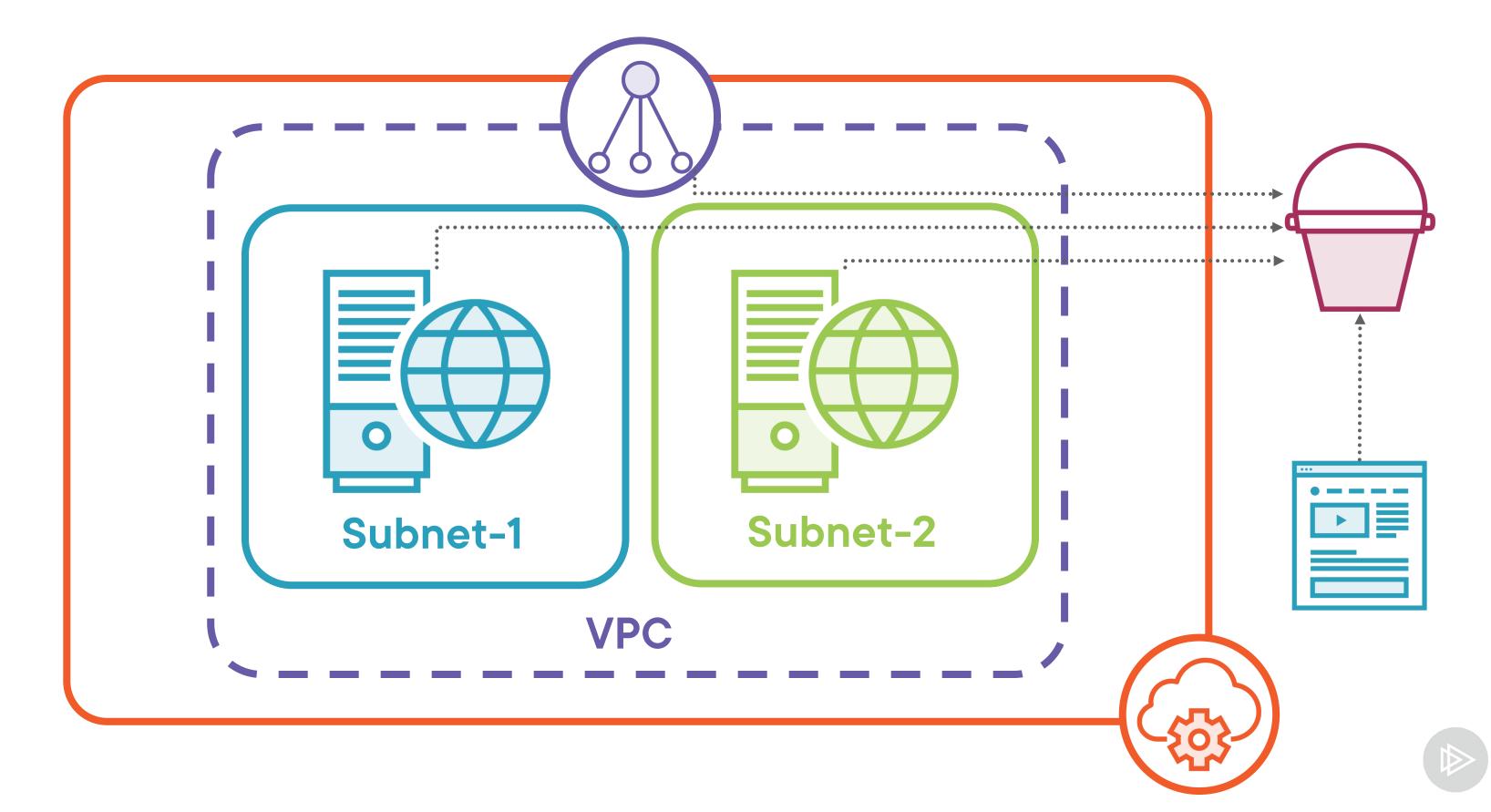
```
Provider Block Syntax
```

```
provider "provider_name" {
  alias = "alias_name"
  # Provider specific arguments
}
```

#### Provider Block Syntax

```
provider "aws" {
 alias = "west"
 # Provider specific arguments
resource "aws_instance" "web_server" {
 provider = aws.west
 # Resource specific arguments
```

# Deployment Architecture



#### S3 and IAM Resources

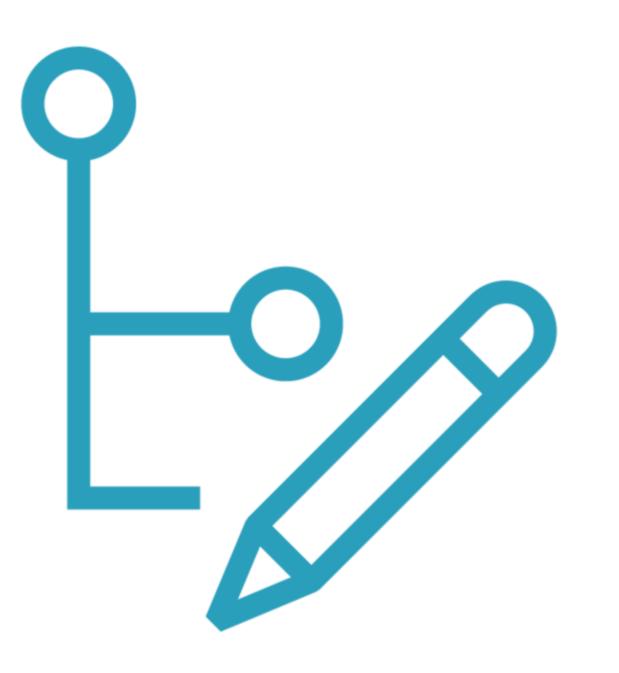
```
# S3 Resources
"aws s3 bucket" # S3 bucket itself
"aws_s3_bucket_object" # Objects in the bucket
# IAM Resources
"aws_iam_role" # Role for instances
"aws_iam_role_policy" # Role policy for S3 access
"aws_iam_instance_profile" # Instance profile
# Data Source
"aws_elb_service_account" # For load balancer access
```



# Planning and Dependencies



## Terraform Planning



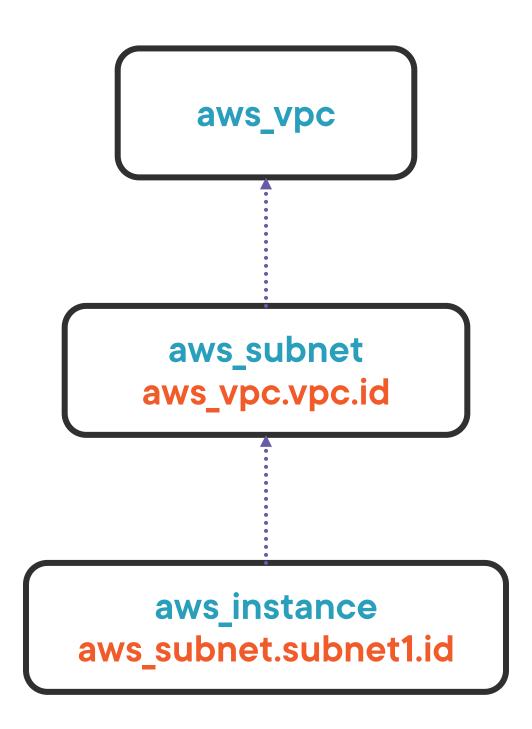
Refresh and inspect state

Dependency graph

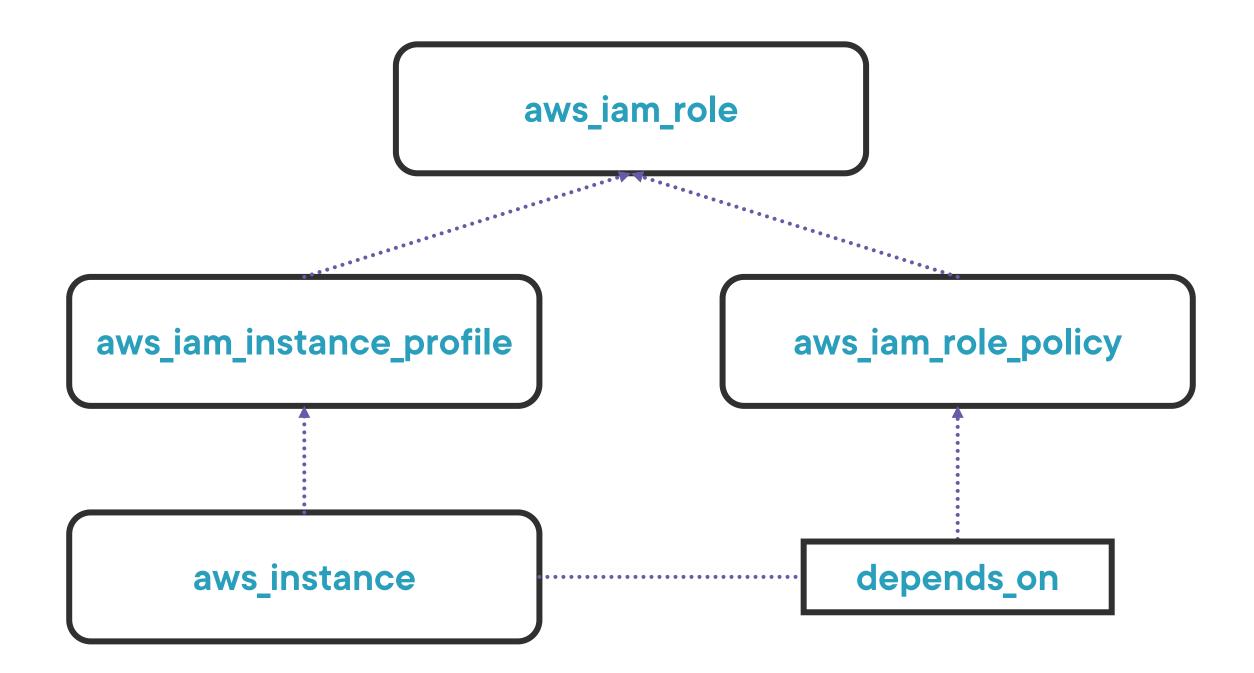
Additions, updates, and deletions

Parallel execution

## Determining Dependencies

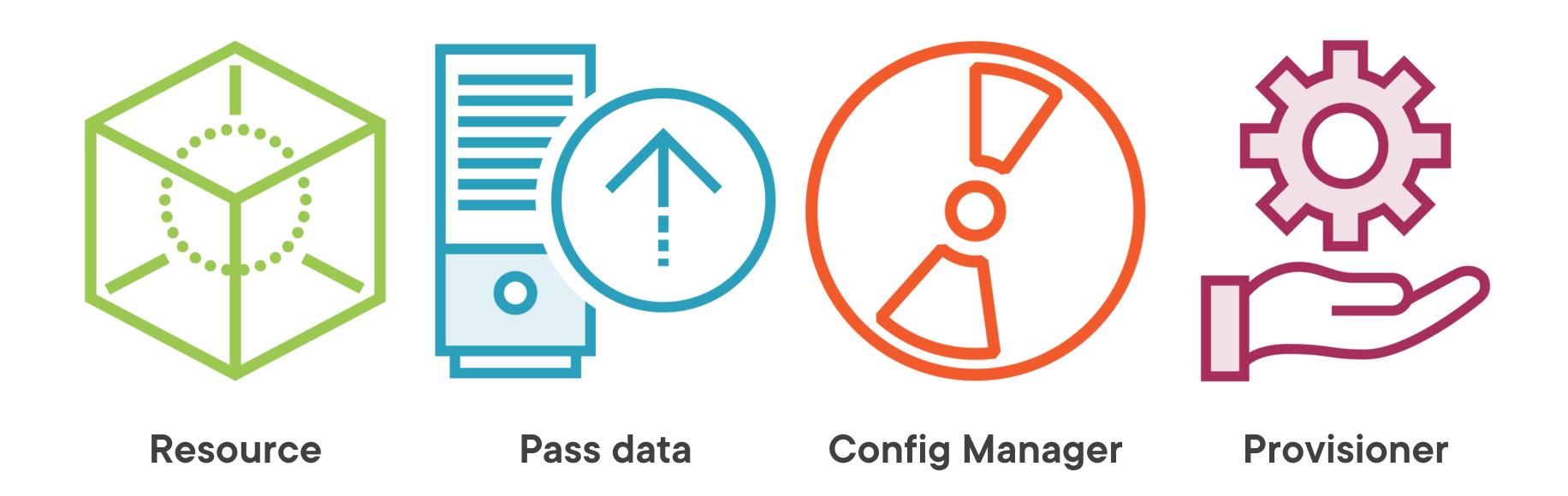


### Determining Dependencies



# Post Deployment Configuration

## Configuration Options



#### Provisioners

**Defined in resource** 

Creation or destruction

Multiple provisioners

null\_resource

Failure options

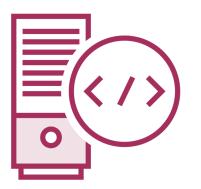
**Last Resort!** 



File



Local-exec



Remote-exec



## Provisioner Example

```
provisioner "file" {
  connection {
    type = "ssh"
    user = "root"
    private_key = var.private_key
    host = self.public_ip
  }
  source = "/local/path/to/file.txt"
  destination = "/path/to/file.txt"
}
```



### Provisioner Example

```
provisioner "local-exec" {
  command = "local command here"
}

provisioner "remote-exec" {
  scripts = ["list", "of", "scripts"]
}
```



#### Summary



Adding a new provider

Specifying provider version and source

Resource dependency

Post deployment configuration



## Up Next:

Using Functions and Looping in Your Configuration