

Adding a New Provider to Your Configuration



Ned Bellavance

HashiCorp Ambassador

@ned1313 | nedinthecloud.com



Module 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

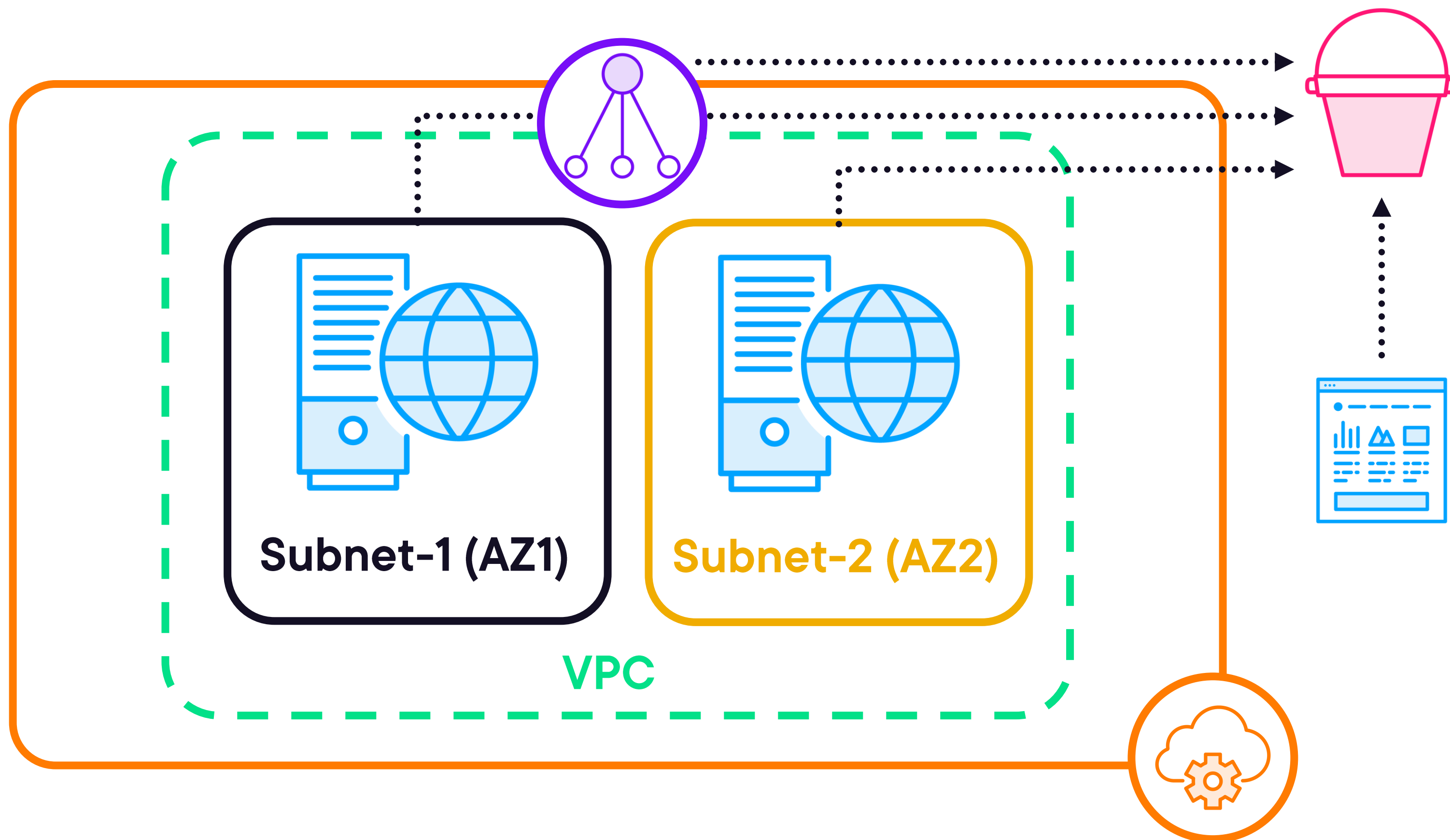
Log traffic to an S3 bucket

Use specific provider versions

Properly format files



Deployment Architecture

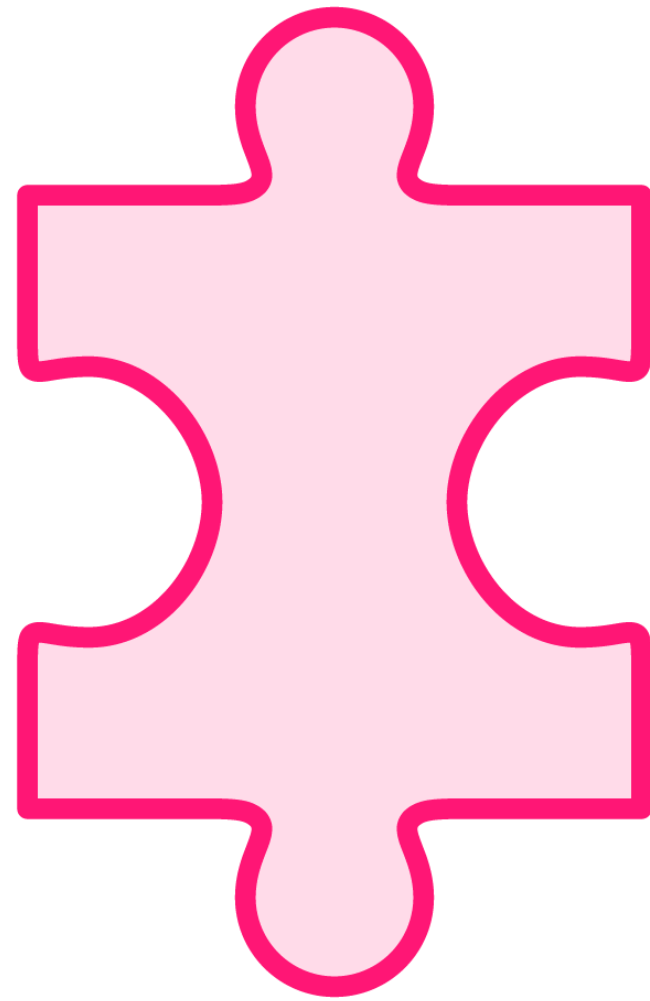




Terraform Providers



Terraform Providers



Public and private registries

Provider tiers:

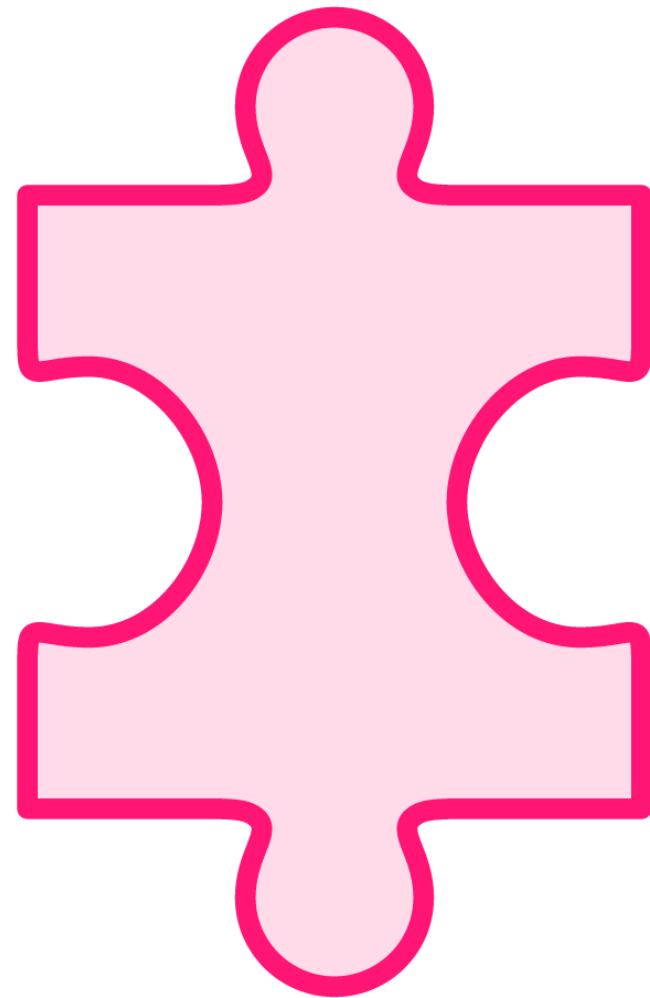
- Official
- Partner
- Community

Open source

Resources and data sources



Terraform Providers



Versioned

- (major).(minor).(patch)
- Version constraints

Multiple instances

- Alias argument



Terraform Block Syntax

terraform.tf

```
terraform {  
    required_version = "version_constraints"  
    required_providers {  
        provider_name = {  
            source = "address_to_provider"  
            version = "version_constraints"  
            #   =, >, <, =>, =<, ~>  
        }  
    }  
}
```



Terraform Block Syntax

terraform.tf

```
terraform {  
    required_version = ">= 1.0,  
    required_providers {  
        aws = {  
            source = "hashicorp/aws"  
            # =, <, >, and ~>  
            version =  
        }  
    }  
}
```



Dependency Lock File

.terraform.lock.hcl

```
# This file is maintained automatically by "terraform init".
# Manual edits may be lost in future updates.

provider "registry.terraform.io/hashicorp/aws" {
  version      = "4.66.1"
  constraints = ">= 4.0"
  hashes = [
    "h1:oi2xfTwb5ZgjA10luE4KThAQF4+1kN0yNbUu5053rsA=",
    "zh:001c707174b7d6bf89a96cf806f925bb852d1a285fb80b81222cbeb4743bcb79",
    "zh:19bc6ac0a7fd1c564fd56c536f1743f71a5e7ca724e21ea51a6a79218939733d",
    "zh:3dac5c27f40b511239e9fe6f97dc0b6c95f630ba328001820ddc764e766a5ca2",
    "zh:49092c92e2565db4cd4c98ec6878386e6957525d3392b63f0d5df4c48a7c1913",
    "zh:4f9e2e1d0c5365a4e6689096cc91ba88ca9c0dc7c633377ba674c1dd856b6a9f",
  ]
}
```





Where Did My Provider Come From?!

main.tf

```
provider "aws" {}  
  
resource "aws_instance" "web_server" {}  
  
data "aws_availability_zones" "available" {}
```



Provider Block Syntax

providers.tf

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



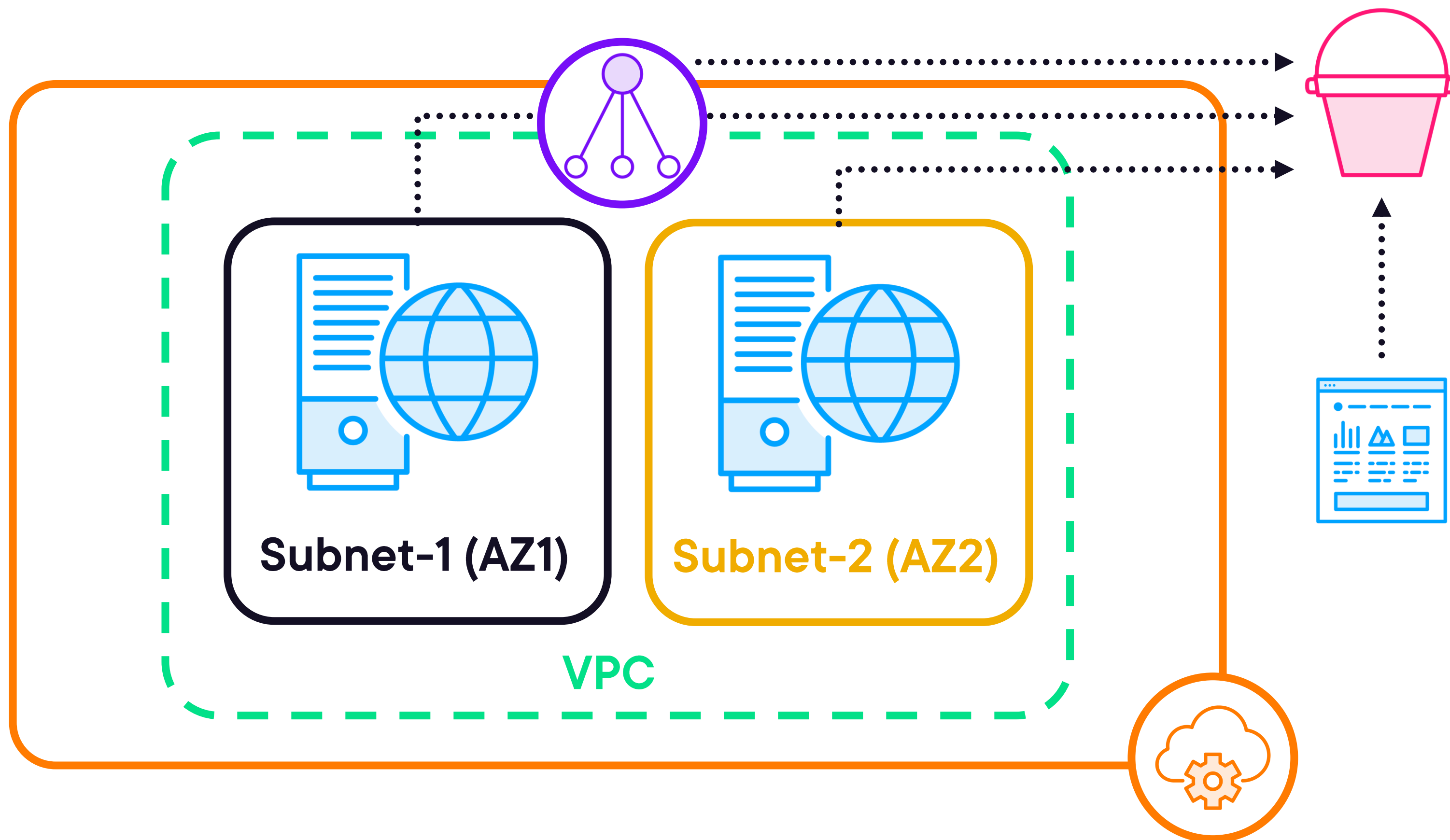
Provider Block Syntax

providers.tf

```
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_object" # Objects in the bucket
```

```
# Instance access
```

```
"aws_iam_role" # Role for instances
```

```
"aws_iam_role_policy" # Role policy for S3 access
```

```
"aws_iam_instance_profile" # Instance profile
```

```
# Load balancer access
```

```
"aws_s3_bucket_policy" # Grant load balancer principal access
```

```
"aws_elb_service_account" # Get load balancer principal id
```

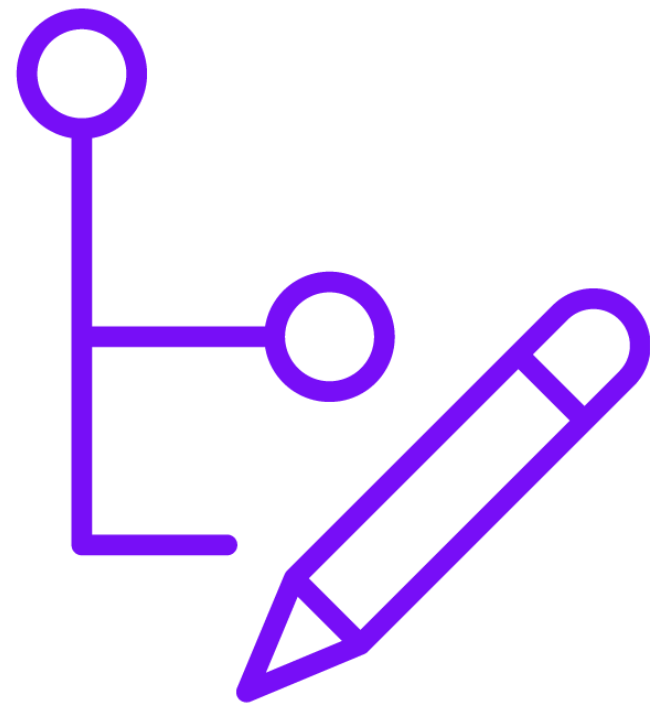




Planning and Dependencies



Terraform Planning



Refresh and inspect state

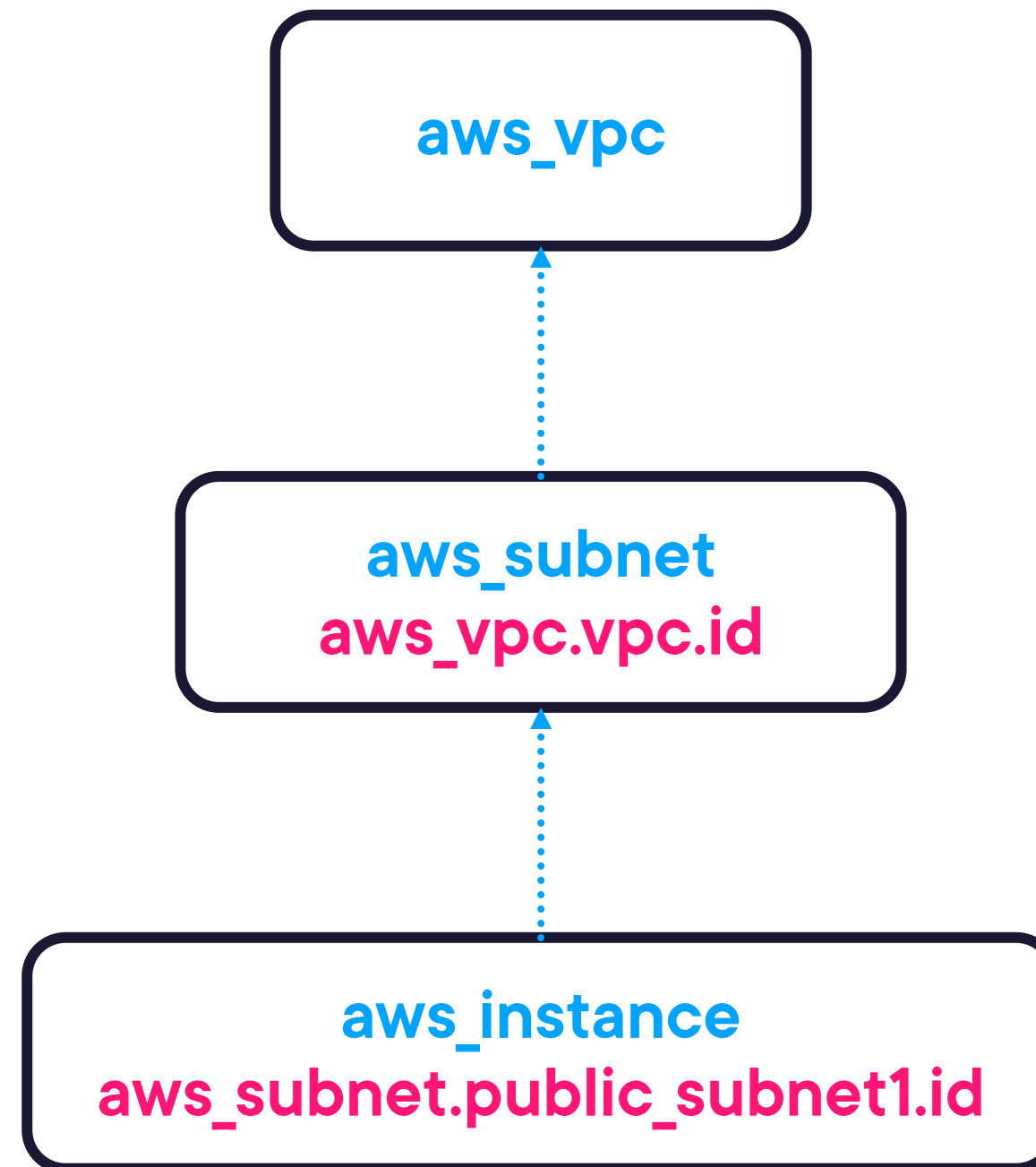
Dependency graph

Additions, updates, and deletions

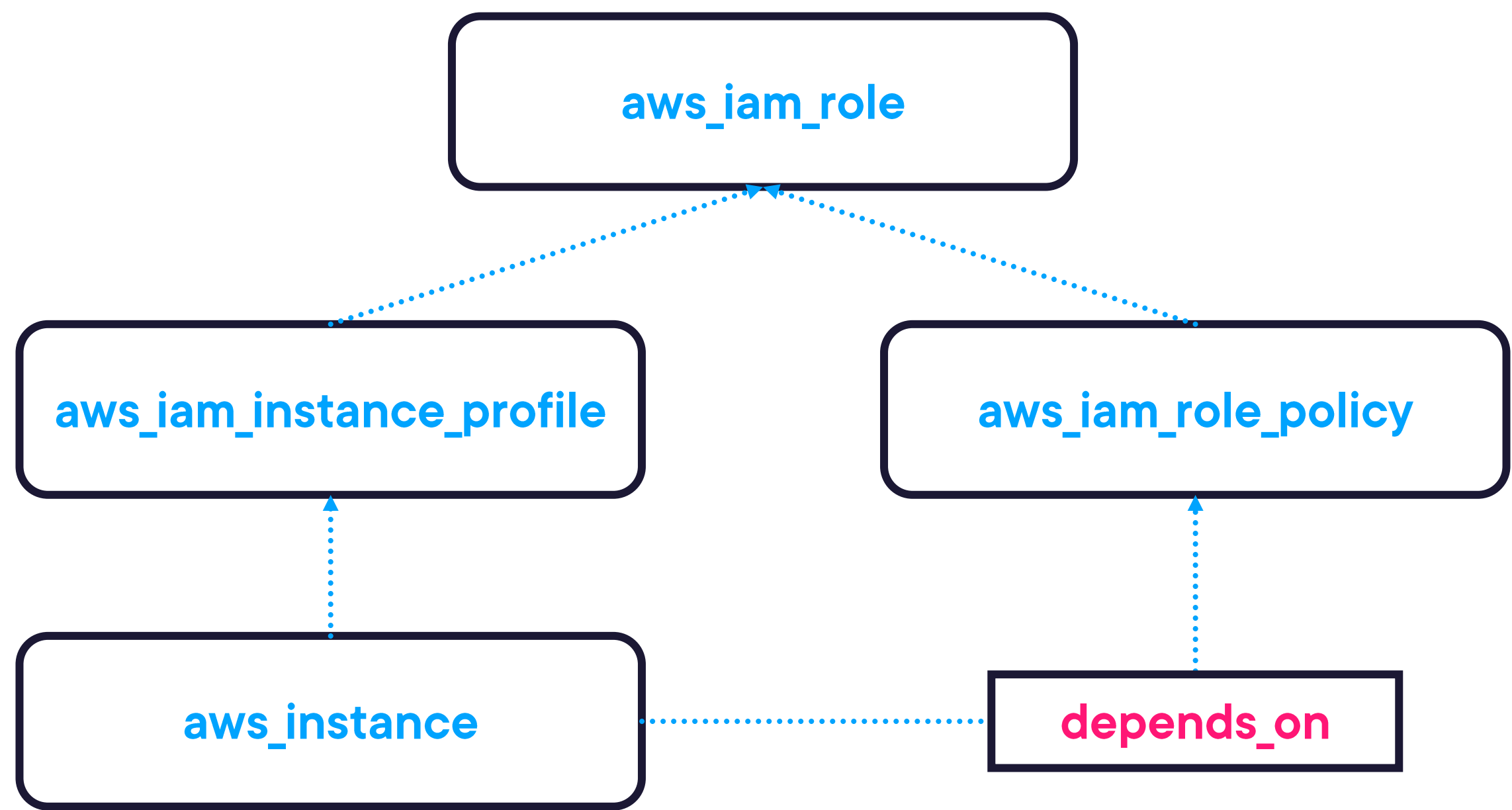
Parallel execution



Determining Dependencies



Determining Dependencies



Meta-argument

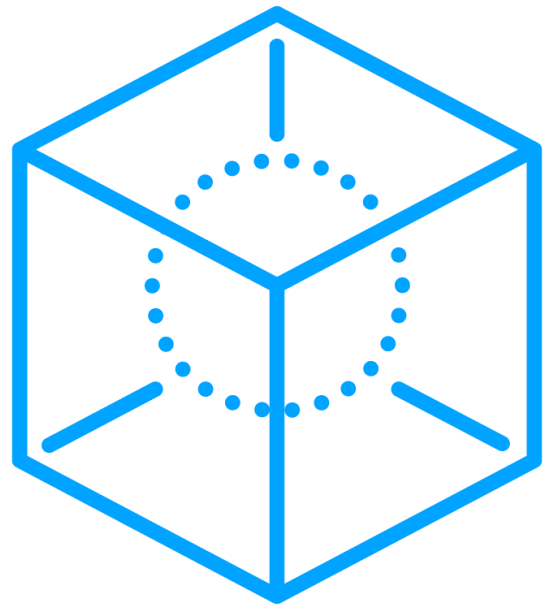
An argument in a resource or data source that instructs Terraform on managing the object and is not used by the provider to configure the target object.



Post Deployment Configuration



Configuration Options



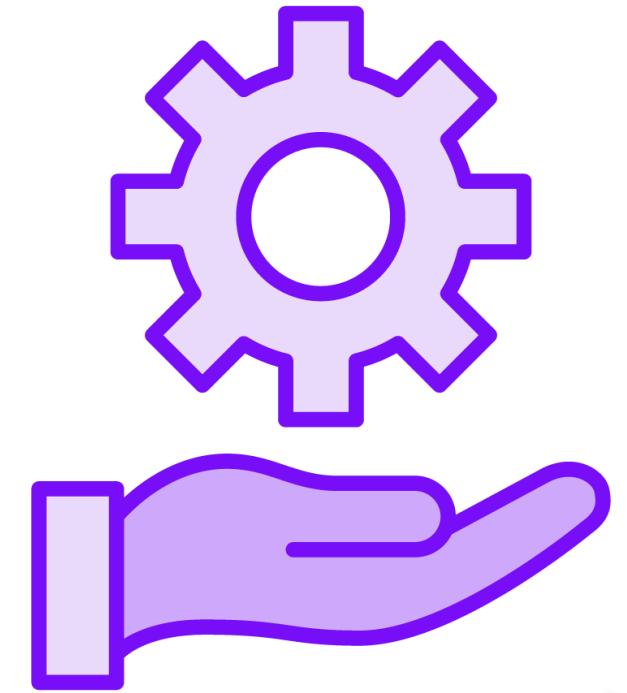
Resource



Pass data

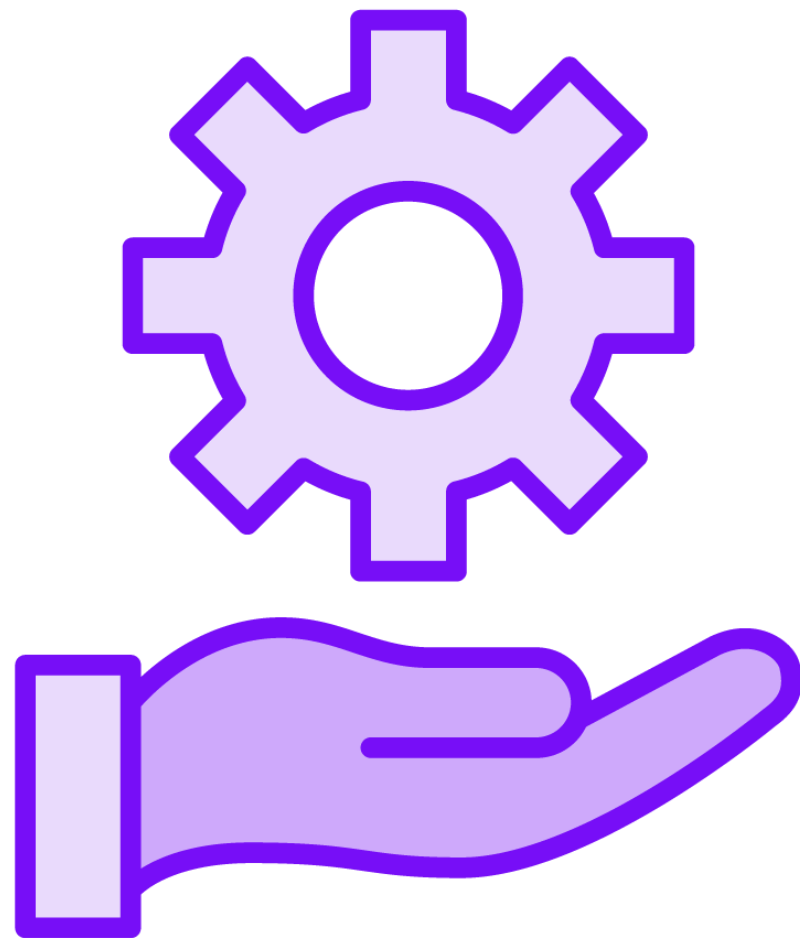


Config manager



Provisioner

Provisioners



Defined in resource

Creation or destruction

Multiple provisioners

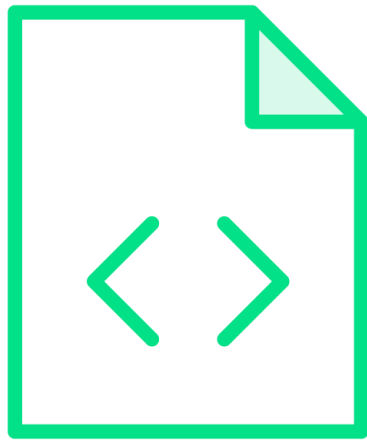
`null_resource` or `terraform_data`

Failure options

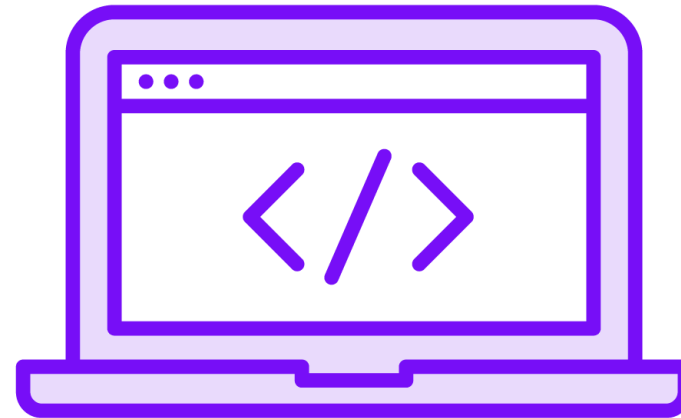
Last Resort!



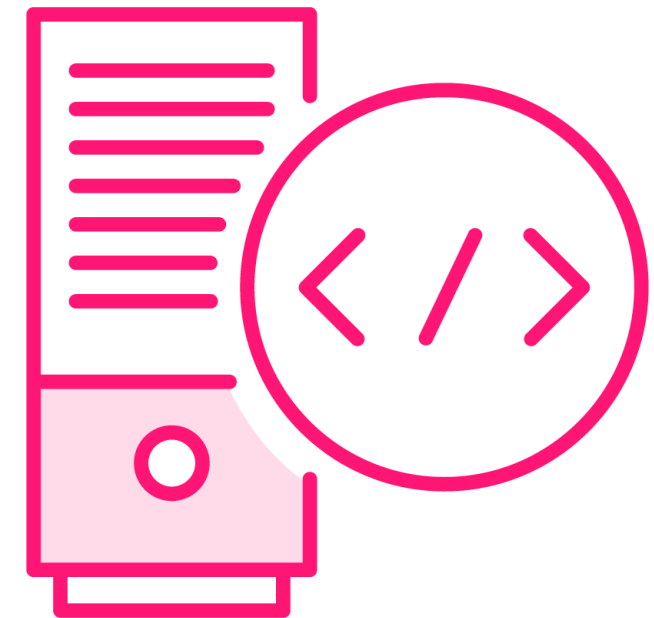
Provisioner Types



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"]  
}
```



Module Summary



Adding a new provider

Specifying provider version and source

Resource dependency

Post deployment configuration



Up Next:

Using Functions and Looping in Your Configuration

