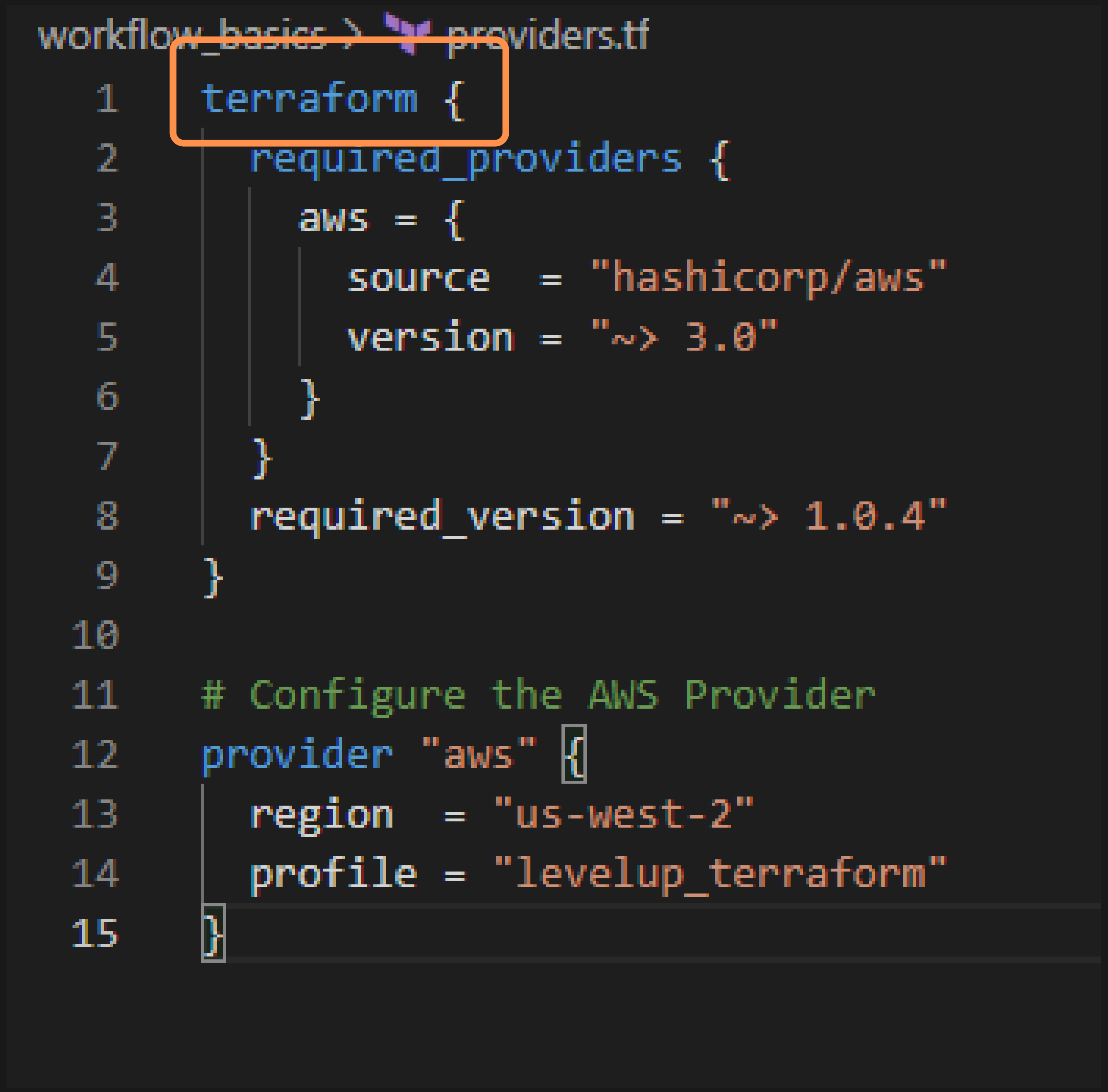
Terraform Provider Configuration

HashiCorp
Terraform

Provider Configuration

- The terraform provider configurations are generally placed in providers.tf
  - Best practice to stick with providers.tf - but not required
  - Also best practice to place the terraform block configuration in providers.tf
- The terraform block contains your global terraform configurations and affects how Terraform behaves
  - Restricting Terraform CLI version
  - Configuring the state backend
  - Enforcing required providers to run your infrastructure environment

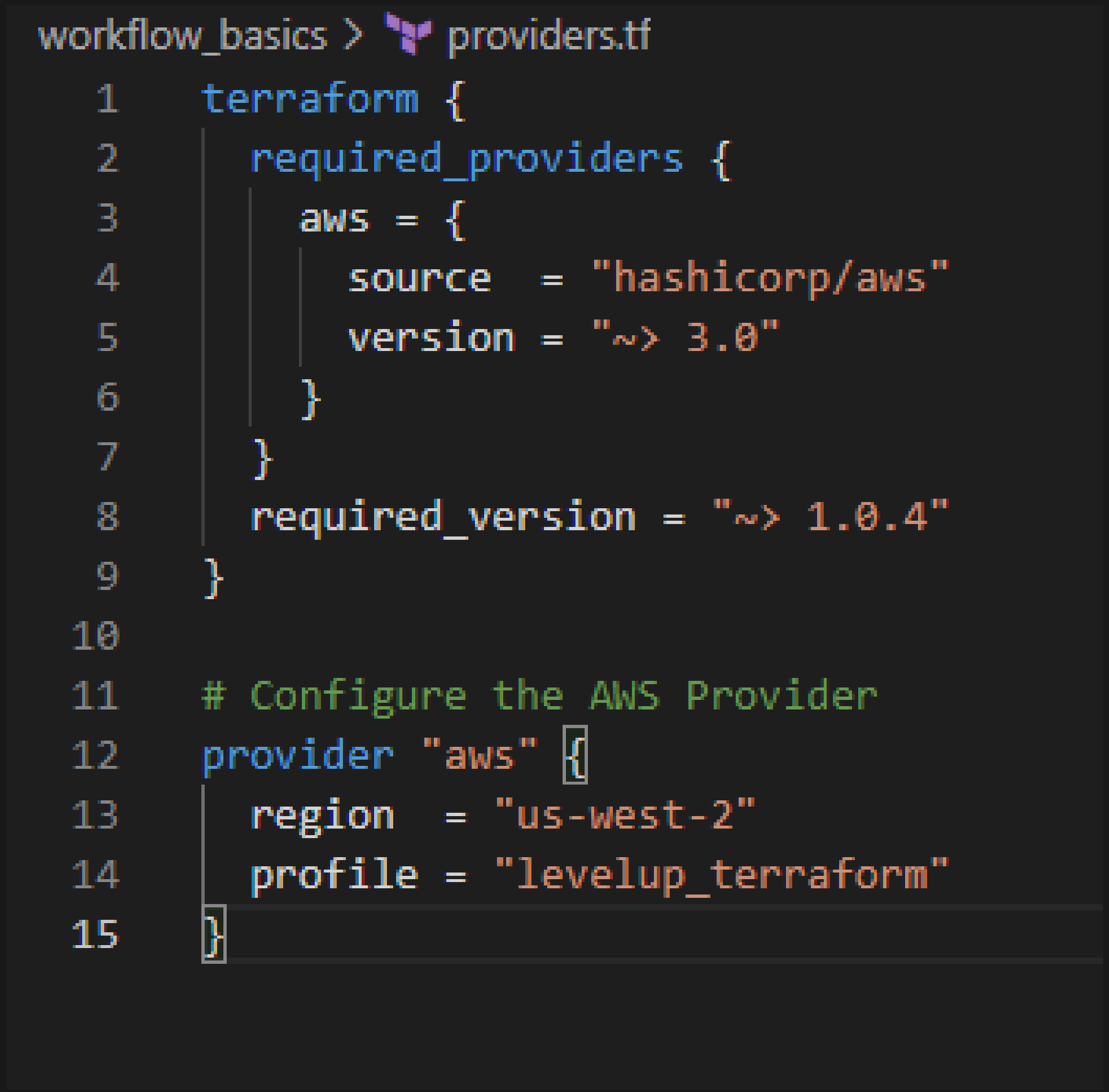


```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
  required_version = "~> 1.0.4"
}

# Configure the AWS Provider
provider "aws" {
  region  = "us-west-2"
  profile = "levelup_terraform"
}
```

Provider Configuration

- required_providers nested block inside terraform block
    - Tells Terraform which providers you need to run your infrastructure
- In example, we require the official AWS provider from HashiCorp, and restrict the version to anything 3.0 and above, but less than 4.0
- We configure additional settings in our AWS provider
    - Giving the region to use, and how to authenticate with AWS
    - In example, we are using us-west-2 region, and the levelup_terraform profile, which is our superuser account

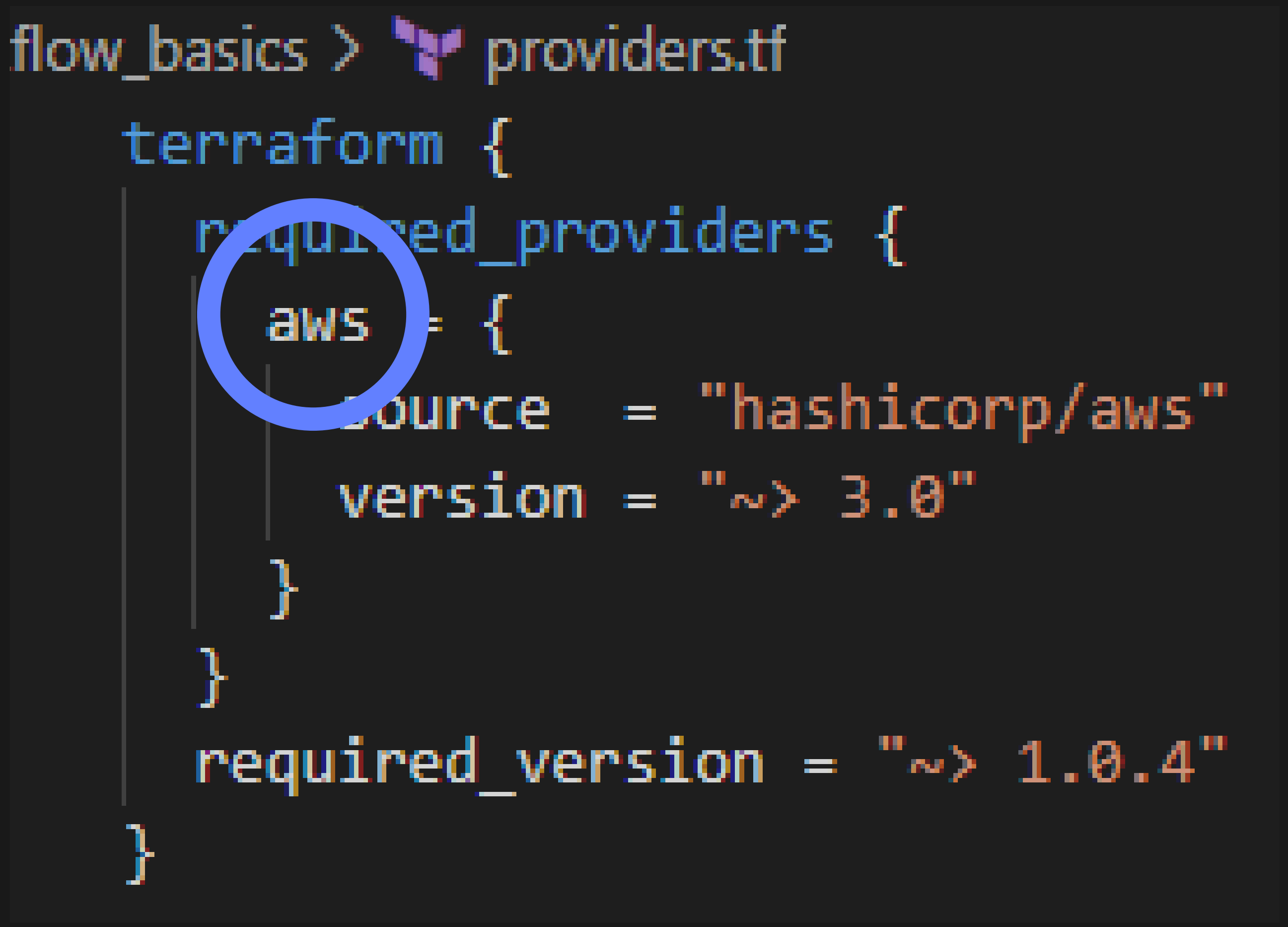


```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
  required_version = "~> 1.0.4"
}

# Configure the AWS Provider
provider "aws" {
  region  = "us-west-2"
  profile = "levelup_terraform"
}
```

Provider Configuration

- The required_providers block consists of three parts:
  - local name - The name given to the declared provider
  - source - Where the provider is from
  - version - The version constraint of the requested provider
- The local name is a unique name assigned to label the provider
- The local name is very important - used as a prefix for all resource objects
  - ec2 instances: aws_instance
  - ami: aws_ami
- Local names help isolate resource types via a namespace
  - Reduces chance of resource name collisions with other Providers

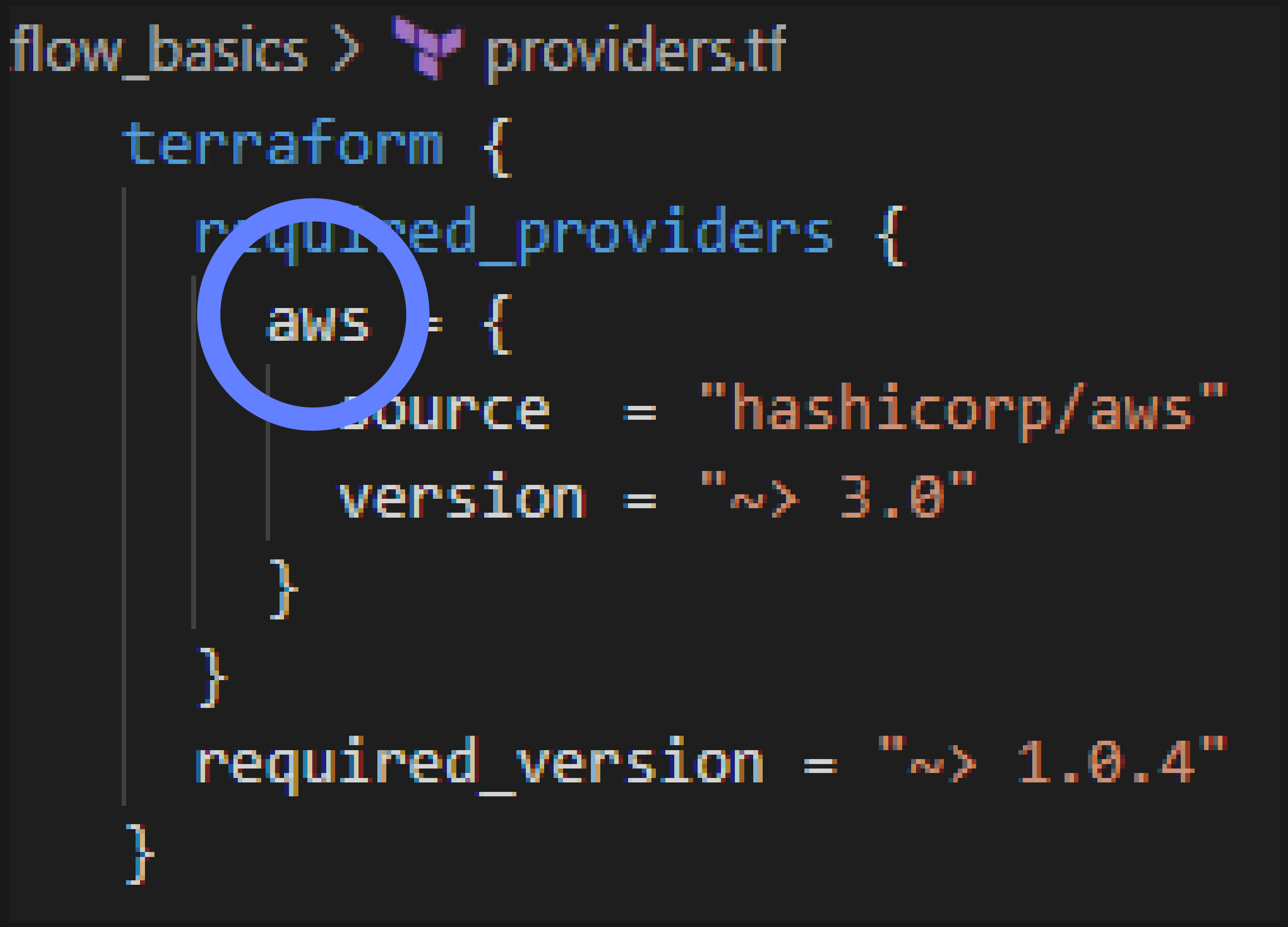


flow_basics > providers.tf

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
  required_version = "~> 1.0.4"
}
```

Provider Configuration

- The local name can be anything, but better to pick the provider preferred name
- Each provider has their own preferred local name
- The most common preferred local names:
  - For AWS provider by HashiCorp - "aws"
  - For GCP provider by HashiCorp - "google"
- You generally would never choose another local name
  - Stick to the provider preferred local name
  - Makes yours and your teams life easier!

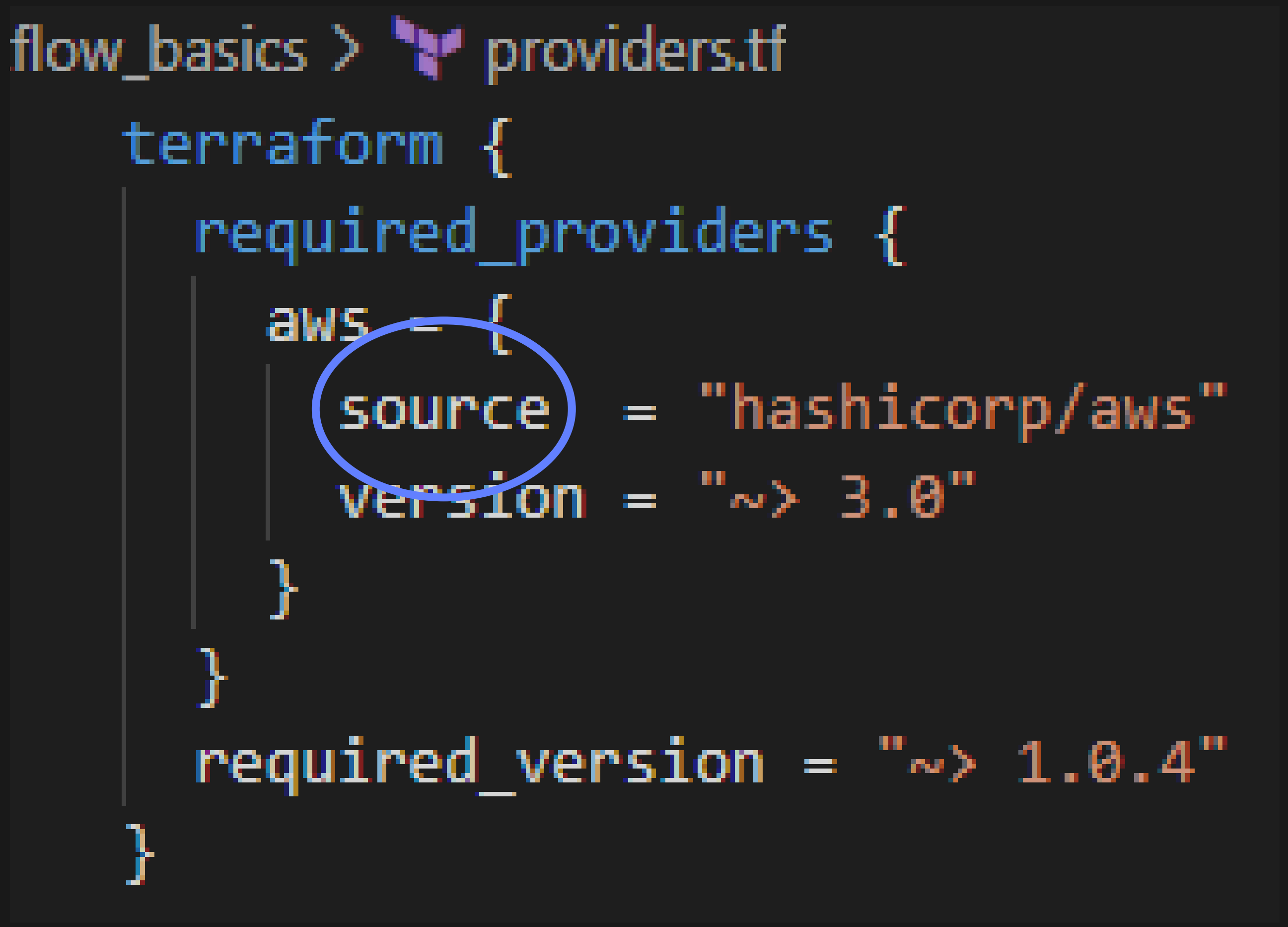


flow_basics > providers.tf

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
  required_version = "~> 1.0.4"
}
```

Provider Configuration

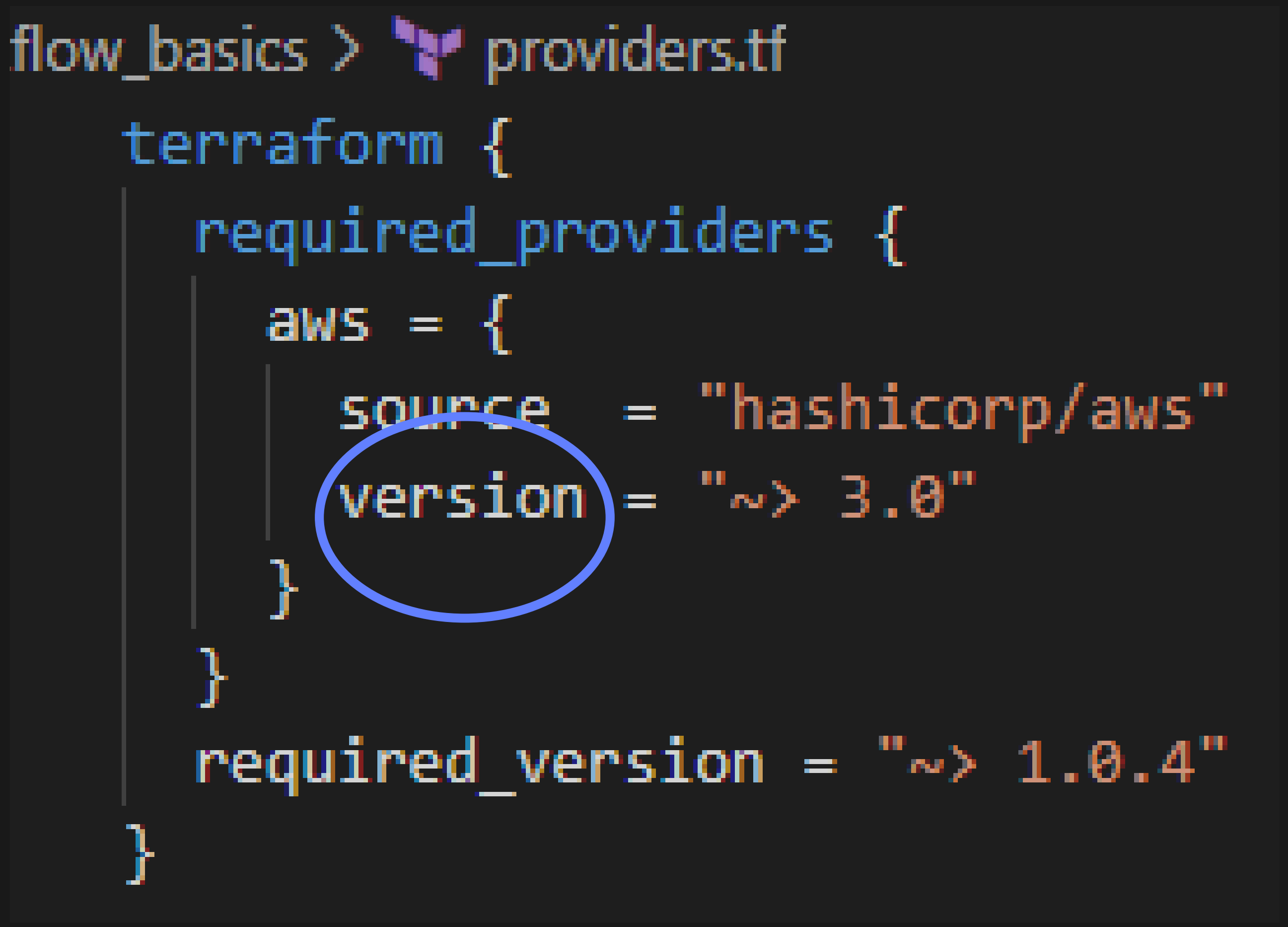
- The source defines where the Provider is being downloaded from
- The value makes up the source address
  - <HOSTNAME>/<NAMESPACE>/<TYPE>
- HOSTNAME
  - The address of the registry
  - If no HOSTNAME is provided, defaults to registry.terraform.io - official public registry
- NAMESPACE
  - The organizational namespace within the registry
  - In this case, the NAMESPACE is hashicorp
- TYPE
  - The unique identifier within the namespace
  - In this case, the TYPE is aws



```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
  required_version = "~> 1.0.4"
}
```

Provider Configuration

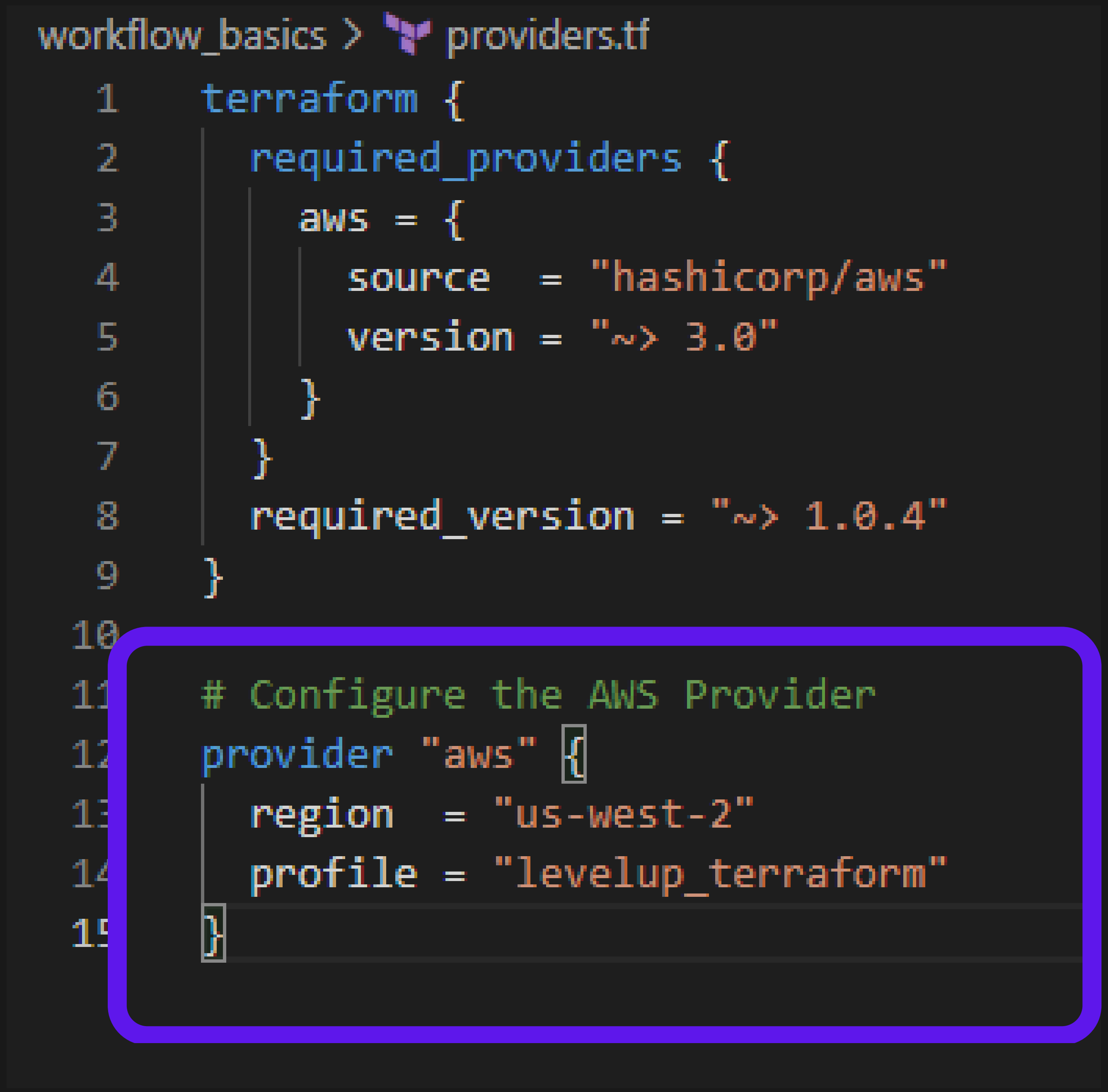
- The version constraint restricts what Provider version is going to be downloaded when running terraform init
  - Each Provider maintains its own version, outside of Terraform
  - Dependency lockfile relies on version constraint to provide version stability
  - Terraform will download the latest version that satisfies the version constraint
- Version constraint is important - ensures everyone in the team downloads the same Provider



```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
  required_version = "~> 1.0.4"
}
```

Provider Configuration

- After initializing the required providers, configure them
  - Provide authentication measures
  - Provide settings to correctly set up your provider
- In context of the AWS Provider
  - Provide the region
  - Provide the AWS profile to authenticate with
- The resource name of the provider is "aws"
  - Must match with the name provided under "required_providers"
- REMEMBER - These configuration blocks should be defined inside a providers.tf file. This is the best practice.



workflow_basics > providers.tf

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
  required_version = "~> 1.0.4"
}

# Configure the AWS Provider
provider "aws" {
  region  = "us-west-2"
  profile = "levelup_terraform"
}
```