8. Infrastructure as Code Using Terraform

1. **Install Terraform**: Download and install the Terraform binary for your operating system from the official Terraform website.

2. **Configure Provider**: Choose the cloud provider you want to work with (e.g., AWS, Azure, Google Cloud) and configure the necessary credentials or authentication methods to access your cloud account.

3. **Create a Terraform Configuration**: Create a new directory for your Terraform project and navigate to it. Inside the directory, create a file named `main.tf` (or any other `.tf` file) to define your infrastructure resources. Use the appropriate provider block for your selected cloud provider and specify resource configurations using Terraform syntax.

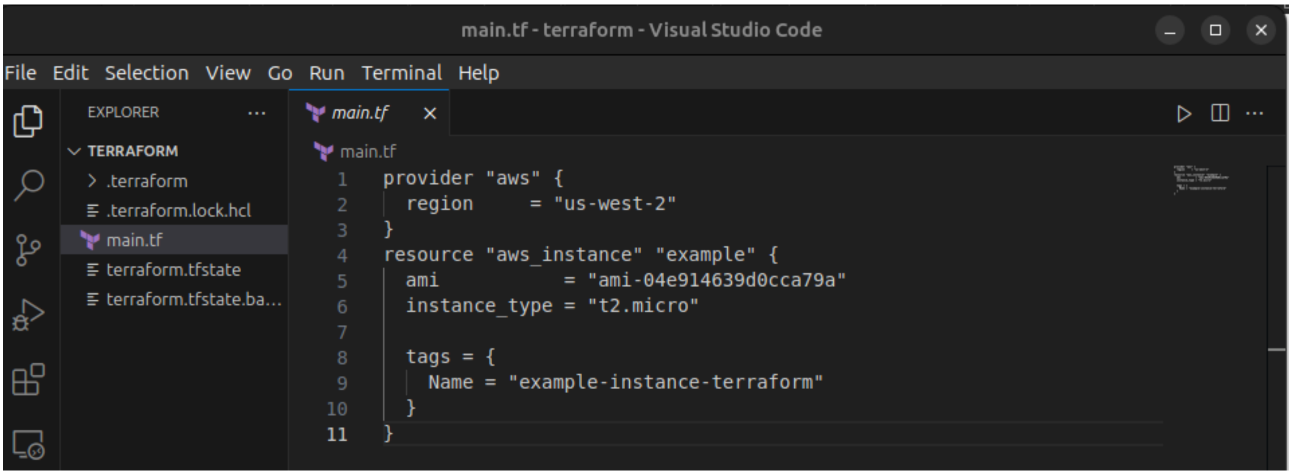
4. **Initialize Terraform**: In the terminal, navigate to your Terraform project directory and run `terraform init`. This command initializes the Terraform working directory, downloads the required provider plugins, and sets up the backend configuration.

5. **Plan and Apply**: Run `terraform plan` to preview the changes that Terraform will make to your infrastructure based on the configuration. Review the plan output to ensure it aligns with your expectations and verifies that the changes are correct. If everything looks good, execute `terraform apply` to apply the changes and provision the infrastructure resources.

6. **Manage Infrastructure**: As your infrastructure needs evolve, update your Terraform configuration file accordingly. Run `terraform plan` to preview the changes, then `terraform apply` to apply those changes to your infrastructure. This enables you to add, modify, or remove resources as required.

7. **Destroy Infrastructure**: If you no longer need the provisioned infrastructure, you can destroy it using `terraform destroy`. This command will remove all the resources created by Terraform based on your configuration. Be cautious when using this command, as it irreversibly deletes your infrastructure.

By following these steps, you can effectively utilize Terraform for Infrastructure as Code, enabling you to provision, manage, and scale your infrastructure resources in a consistent and automated manner.



```
provider "aws" {
  region     = "us-west-2"
}
resource "aws_instance" "example" {
  ami           = "ami-04e914639d0cca79a"
  instance_type = "t2.micro"

  tags = {
    Name = "example-instance-terraform"
  }
}
```