Terraform

Commands:
wget
https://releases.hashicorp.com/terraform/0.12.29/terraform_0.12.29_linux_amd64.zip

sudo apt install unzip

unzip *

mv terraform /usr/local/bin/

terraform -help

apt-get install aws

Then configure AWS credentials:
Create a user in AWS IAM and grant it with required privilege
Create a credentials file in ~/.aws/credentials:

Create a config file in ~/.aws/config:

```
[default]
region=ap-southeast-2
output=json
```

Create example.tf

```
provider "aws" {
  profile = "default"
  region  = "ap-southeast-2"
}

variable "subnet_id" {}

data "aws_subnet" "selected" {
  id = "${var.subnet_id}"
}

resource "aws_instance" "example" {
  ami           = "ami-0a58e22c727337c51"
  instance_type = "t2.micro"
  subnet_id     = data.aws_subnet.selected.id
  private_ip    = "10.0.1.8"
  tags          = {
     Name = "Terraform Modified Instance"
  }
}


resource "aws_eip" "instance-ip" {
  vpc                       = true
  instance                  = aws_instance.example.id
  associate_with_private_ip = "10.0.1.8"
  depends_on                = [aws_instance.example]
}
```

Then:
terraform init // initialize a working directory with Terraform configuration files
terraform validate

terraform apply

```
ubuntu@ip-10-0-1-183:~/files/terraform/learn-terraform-aws-instance$ terraform apply
var.subnet_id
  Enter a value: subnet-08[illegible]e0

data.aws_subnet.selected: Refreshing state...

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_eip.instance-ip will be created
  + resource "aws_eip" "instance-ip" {
      + allocation_id             = (known after apply)
      + associate_with_private_ip = "10.0.1.8"
      + association_id            = (known after apply)
      + customer_owned_ip         = (known after apply)
      + domain                    = (known after apply)
      + id                        = (known after apply)
      + instance                  = (known after apply)
      + network_interface         = (known after apply)
      + private_dns               = (known after apply)
      + private_ip                = (known after apply)
      + public_dns                = (known after apply)
      + public_ip                 = (known after apply)
      + public_ipv4_pool          = (known after apply)
      + vpc                       = true
    }

  # aws_instance.example will be created
  + resource "aws_instance" "example" {
      + ami                          = "ami-0a58e22c727337c51"
      + arn                          = (known after apply)
      + associate_public_ip_address  = (known after apply)
      + availability_zone            = (known after apply)
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.micro"
      + ipv6_address_count           = (known after apply)
      + ipv6_addresses               = (known after apply)
      + key_name                     = (known after apply)
      + network_interface_id         = (known after apply)
      + outpost_arn                  = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (known after apply)
      + primary_network_interface_id = (known after apply)
      + private_dns                  = (known after apply)
      + private_ip                   = "10.0.1.8"
      + public_dns                   = (known after apply)
      + public_ip                    = (known after apply)
      + security_groups              = (known after apply)
      + source_dest_check            = true
      + subnet_id                    = "subnet-08c5c9b9531d599e0"
      + tags                         = {
          + "Name" = "Terraform Modified Instance"
```

```
Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Creation complete after 20s [id=i-0237b    .1    a    ]
aws_eip.instance-ip: Creating...
aws_eip.instance-ip: Creation complete after 1s [id=eipalloc-8     -e          .97]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

Now we have the modified AWS EC2 instance

terraform destroy
terminates the resource