QA) Design a set of Terraform configurations to establish the following AWS resources. Organize the primary code in `main.tf`, declare variables in `variables.tf`, and assign values in `custom.tfvars` for:

1. EC2 Instances
2. SSH Key Pairs
3. Security Groups

==========

SOLUTION :

The primary code in `main.tf`:-

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.33.0"
    }
  }
}

provider "aws" {
  # configuration options
}

resource "aws_key_pair" "sim_ssh_key" {
 key_name   = var.SSH_KEY_NAME
```

```
  public_key = var.SSH_PUB_KEY
}

resource "aws_instance" "terraform_inst" {
  ami           = var.AMI_ID
  instance_type = var.INST_TYPE
  tags = {
    Name = var.EC2_TAG
  }
}

resource "aws_security_group" "sim_terra_sg" {
    name = var.SG_NAME

 ingress {
    from_port  = var.HTTP_PORT
    to_port    = var.HTTP_PORT
    protocol   = "tcp"
    cidr_blocks = [var.CIDR_RANGE]
  }

 ingress {
    from_port  = var.HTTPS_PORT
    to_port    = var.HTTPS_PORT
    protocol   = "tcp"
```

```
    cidr_blocks = [var.CIDR_RANGE]

  }

}
```

The CLI:-

```
root@DESKTOP-NJSOG33:TERRAFORM# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.33.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```
 ------------------------------

 Declared variables in `variables.tf:-

 -------------------------------

 #THIS IS FOR EC2

variable  "AMI_ID" {

   type = string

}


variable  "INST_TYPE" {

   type = string

}


variable  "EC2_TAG" {

   type = string

}


#THIS IS FOR SSH KEY

variable  "SSH_KEY_NAME" {

   type = string

}


variable  "SSH_PUB_KEY" {

   type = string
```

```
}

#THIS IS FOR SECURITY GROUP

variable  "SG_NAME" {
   type = string
}

variable  "HTTP_PORT" {
   type = number
}

variable  "HTTPS_PORT" {
   type = number
}

variable "CIDR_RANGE" {
   type = string
}
```

On CLI :-

root@DESKTOP-NJSOG33:TERRAFORM# terraform plan -var-file=custom.tfvars

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the

following symbols:

  + create

Terraform will perform the following actions:

  # aws_instance.terraform_inst will be created

  + resource "aws_instance" "terraform_inst" {

      + ami   = "ami-00952f27cf14db9cd"

      + 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)

      + disable_api_stop   = (known after apply)

      + disable_api_termination   = (known after apply)

      + ebs_optimized   = (known after apply)

      + get_password_data   = false

      + host_id   = (known after apply)

      + host_resource_group_arn   = (known after apply)

      + iam_instance_profile   = (known after apply)

      + id   = (known after apply)

      + instance_initiated_shutdown_behavior = (known after apply)

      + instance_lifecycle   = (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)
+ monitoring                        = (known after apply)
+ outpost_arn                       = (known after apply)
+ password_data                     = (known after apply)
+ placement_group                   = (known after apply)
+ placement_partition_number        = (known after apply)
+ primary_network_interface_id      = (known after apply)
+ private_dns                       = (known after apply)
+ private_ip                        = (known after apply)
+ public_dns                        = (known after apply)
+ public_ip                         = (known after apply)
+ secondary_private_ips             = (known after apply)
+ security_groups                   = (known after apply)
+ source_dest_check                 = true
+ spot_instance_request_id          = (known after apply)
+ subnet_id                         = (known after apply)
+ tags                              = {
    + "Name" = "Hello-World"
  }
+ tags_all                          = {
    + "Name" = "Hello-World"
  }
+ tenancy                           = (known after apply)
```

```
    + user_data                     = (known after apply)
    + user_data_base64              = (known after apply)
    + user_data_replace_on_change   = false
    + vpc_security_group_ids        = (known after apply)
  }

 # aws_key_pair.sim_ssh_key will be created
 + resource "aws_key_pair" "sim_ssh_key" {
    + arn           = (known after apply)
    + fingerprint   = (known after apply)
    + id            = (known after apply)
    + key_name      = "terra.sim.key"
    + key_name_prefix = (known after apply)
    + key_pair_id   = (known after apply)
    + key_type      = (known after apply)
    + public_key    = "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDI1OilC6Ljn/JfviO7VFXoof25l0kXT9NP1+l4Ak8wgqD4GUY3Gnqx
sADYCe+9emQmORMnQaA3V5zj70flkKyXsZsDP9lVLlkgMJDGefcL8VoaN4u+Sxn0qagW8QsQhIFK+pFZr9y
mM4oBIAD8vBcxXYMDzaAvYsj1ZauoZNZRJ/wmY2rnlULD+Zvjw6ips6FH9oRqAgXjMDlWJtTSyRquQunWM
8vuVTd+NbRrY/dFN86HpetYEJAWIeP8EtwfoIUTBTjz9lKoVKLyQNXOGn/x3AFgR167mmXb0li37Dj9BtPJ1z7
31NWOcJqN+tFo98XCLtjU0DWVqEs2fSNWtRPN03pOiYOXRuz6VOzXN7OFsFUkGM4uTG6Sd4YHaCHIaG5
MVIMDG17Fq5jJskqQkZwp6oddVCwAI1nTW6n35mH3/ZNRA3QSnAqcRmoJZE6y3me7Vp4T8Lf03LZIapTI/
HDCP8uYG98NN9hGIGuWwPvlc8ZxSsoazrT0uY6gFrh/kPk="
    + tags_all      = (known after apply)
  }

 # aws_security_group.sim_terra_sg will be created
 + resource "aws_security_group" "sim_terra_sg" {
```

```
+ arn                  = (known after apply)
+ description          = "Managed by Terraform"
+ egress               = (known after apply)
+ id                   = (known after apply)
+ ingress              = [
  + {
    + cidr_blocks      = [
      + "0.0.0.0/0",
      ]
    + description      = ""
    + from_port        = 443
    + ipv6_cidr_blocks = []
    + prefix_list_ids  = []
    + protocol         = "tcp"
    + security_groups  = []
    + self             = false
    + to_port          = 443
    },
  + {
    + cidr_blocks      = [
      + "0.0.0.0/0",
      ]
    + description      = ""
    + from_port        = 80
    + ipv6_cidr_blocks = []
```

```
      + prefix_list_ids  = []
      + protocol        = "tcp"
      + security_groups  = []
      + self            = false
      + to_port         = 80
    },
  ]
  + name                = "MY-SG01"
  + name_prefix         = (known after apply)
  + owner_id            = (known after apply)
  + revoke_rules_on_delete = false
  + tags_all            = (known after apply)
  + vpc_id              = (known after apply)
}

Plan: 3 to add, 0 to change, 0 to destroy.

─────────────────────────────────────────────────────────────────────────────
──────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if

you run "terraform apply" now.

---------------
```

assign values in `custom.tfvars`

#THIS IS FOR EC2

AMI_ID = "ami-00952f27cf14db9cd"

INST_TYPE = "t2.micro"

EC2_TAG = {Name = "Hello-World"}

#THIS IS FOR SSH KEY

SSH_KEY_NAME = "terra.sim.key"

SSH_PUB_KEY = "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDI1OilC6Ljn/JfviO7VFXoof25l0kXT9NP1+l4Ak8wgqD4GUY3Gnqx
sADYCe+9emQmORMnQaA3V5zj70flkKyXsZsDP9lVLlkgMJDGefcL8VoaN4u+Sxn0qagW8QsQhIFK+pFZr9y
mM4oBIAD8vBcxXYMDzaAvYsj1ZauoZNZRJ/wmY2rnlULD+Zvjw6ips6FH9oRqAgXjMDlWJtTSyRquQunWM
8vuVTd+NbRrY/dFN86HpetYEJAWIeP8EtwfoIUTBTjz9lKoVKLyQNXOGn/x3AFgR167mmXb0li37Dj9BtPJ1z7
31NWOcJqN+tFo98XCLtjU0DWVqEs2fSNWtRPN03pOiYOXRuz6VOzXN7OFsFUkGM4uTG6Sd4YHaCHIaG5
MVIMDG17Fq5jJskqQkZwp6oddVCwAI1nTW6n35mH3/ZNRA3QSnAqcRmoJZE6y3me7Vp4T8Lf03LZIapTI/
HDCP8uYG98NN9hGIGuWwPvlc8ZxSsoazrT0uY6gFrh/kP="

#THIS IS FOR SECURITY GROUP

SG_NAME = "MY-SG01"

HTTP_PORT = 80

HTTPS_PORT = 443

CIDR_RANGE = "0.0.0.0/0"

On CLI:-

root@DESKTOP-NJSOG33:TERRAFORM# terraform apply -var-file=custom.tfvars -auto-approve

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the

following symbols:

  + create

Terraform will perform the following actions:

  # aws_instance.terraform_inst will be created

  + resource "aws_instance" "terraform_inst" {

      + ami   = "ami-00952f27cf14db9cd"

      + 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)

      + disable_api_stop   = (known after apply)

      + disable_api_termination   = (known after apply)

      + ebs_optimized   = (known after apply)

      + get_password_data   = false

      + host_id   = (known after apply)

      + host_resource_group_arn   = (known after apply)

      + iam_instance_profile   = (known after apply)

      + id   = (known after apply)

      + instance_initiated_shutdown_behavior = (known after apply)

```
+ instance_lifecycle               = (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)
+ monitoring                       = (known after apply)
+ outpost_arn                      = (known after apply)
+ password_data                    = (known after apply)
+ placement_group                  = (known after apply)
+ placement_partition_number       = (known after apply)
+ primary_network_interface_id     = (known after apply)
+ private_dns                      = (known after apply)
+ private_ip                       = (known after apply)
+ public_dns                       = (known after apply)
+ public_ip                        = (known after apply)
+ secondary_private_ips            = (known after apply)
+ security_groups                  = (known after apply)
+ source_dest_check                = true
+ spot_instance_request_id         = (known after apply)
+ subnet_id                        = (known after apply)
+ tags                             = {
    + "Name" = "Hello-World"
  }
+ tags_all                         = {
```

```
      + "Name" = "Hello-World"
    }
  + tenancy                     = (known after apply)
  + user_data                   = (known after apply)
  + user_data_base64            = (known after apply)
  + user_data_replace_on_change = false
  + vpc_security_group_ids      = (known after apply)
 }

# aws_key_pair.sim_ssh_key will be created
+ resource "aws_key_pair" "sim_ssh_key" {
  + arn           = (known after apply)
  + fingerprint   = (known after apply)
  + id            = (known after apply)
  + key_name      = "terra.sim.key"
  + key_name_prefix = (known after apply)
  + key_pair_id   = (known after apply)
  + key_type      = (known after apply)
  + public_key    = "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDI1OilC6Ljn/JfviO7VFXoof25l0kXT9NP1+l4Ak8wgqD4GUY3Gnqx
sADYCe+9emQmORMnQaA3V5zj70flkKyXsZsDP9lVLlkgMJDGefcL8VoaN4u+Sxn0qagW8QsQhIFK+pFZr9y
mM4oBIAD8vBcxXYMDzaAvYsj1ZauoZNZRJ/wmY2rnlULD+Zvjw6ips6FH9oRqAgXjMDlWJtTSyRquQunWM
8vuVTd+NbRrY/dFN86HpetYEJAWIeP8EtwfoIUTBTjz9lKoVKLyQNXOGn/x3AFgR167mmXb0li37Dj9BtPJ1z7
31NWOcJqN+tFo98XCLtjU0DWVqEs2fSNWtRPN03pOiYOXRuz6VOzXN7OFsFUkGM4uTG6Sd4YHaCHIaG5
MVIMDG17Fq5jJskqQkZwp6oddVCwAI1nTW6n35mH3/ZNRA3QSnAqcRmoJZE6y3me7Vp4T8Lf03LZIapTI/
HDCP8uYG98NN9hGIGuWwPvlc8ZxSsoazrT0uY6gFrh/kPk="
  + tags_all      = (known after apply)
 }
```

```
# aws_security_group.sim_terra_sg will be created
+ resource "aws_security_group" "sim_terra_sg" {
    + arn                    = (known after apply)
    + description            = "Managed by Terraform"
    + egress                 = (known after apply)
    + id                     = (known after apply)
    + ingress                = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = ""
            + from_port        = 443
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 443
          },
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
```

```
            + description      = ""
            + from_port        = 80
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 80
          },
      ]
    + name                 = "MY-SG01"
    + name_prefix          = (known after apply)
    + owner_id             = (known after apply)
    + revoke_rules_on_delete = false
    + tags_all             = (known after apply)
    + vpc_id               = (known after apply)
  }

Plan: 3 to add, 0 to change, 0 to destroy.
aws_key_pair.sim_ssh_key: Creating...
aws_security_group.sim_terra_sg: Creating...
aws_instance.terraform_inst: Creating...
aws_key_pair.sim_ssh_key: Creation complete after 1s [id=terra.sim.key]
aws_security_group.sim_terra_sg: Creation complete after 2s [id=sg-0149a6c14e0c85d50]
aws_instance.terraform_inst: Still creating... [10s elapsed]
```

aws_instance.terraform_inst: Still creating... [20s elapsed]

aws_instance.terraform_inst: Still creating... [30s elapsed]

aws_instance.terraform_inst: Creation complete after 32s [id=i-0c5c707e2b60b8630]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

root@DESKTOP-NJSOG33:TERRAFORM#

NOTE :-

Once your operation is completed, you can destroy all the recent changes and creation on AWS using CLI :-

root@DESKTOP-NJSOG33:TERRAFORM# terraform destroy -var-file=custom.tfvars

aws_key_pair.sim_ssh_key: Refreshing state... [id=terra.sim.key]

aws_security_group.sim_terra_sg: Refreshing state... [id=sg-04455219d59c2cdbe]

aws_instance.terraform_inst: Refreshing state... [id=i-0a29e681c775bdc97]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the

following symbols:

- destroy

Terraform will perform the following actions:

```
  # aws_instance.terraform_inst will be destroyed
  - resource "aws_instance" "terraform_inst" {
      - ami                          = "ami-00952f27cf14db9cd" -> null
      - arn                          = "arn:aws:ec2:ap-south-1:043241213129:instance/i-0a29e681c775bdc97" ->
null
      - associate_public_ip_address        = true -> null
      - availability_zone                  = "ap-south-1a" -> null
      - cpu_core_count                      = 1 -> null
      - cpu_threads_per_core                = 1 -> null
      - disable_api_stop                    = false -> null
      - disable_api_termination             = false -> null
      - ebs_optimized                       = false -> null
      - get_password_data                   = false -> null
      - hibernation                        = false -> null
      - id                                 = "i-0a29e681c775bdc97" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state                      = "running" -> null
      - instance_type                       = "t2.micro" -> null
      - ipv6_address_count                  = 0 -> null
      - ipv6_addresses                      = [] -> null
      - monitoring                         = false -> null
      - placement_partition_number          = 0 -> null
      - primary_network_interface_id        = "eni-0df7024948accd26e" -> null
      - private_dns                         = "ip-172-31-40-107.ap-south-1.compute.internal" -> null
```

- private_ip = "172.31.40.107" -> null

- public_dns = "ec2-13-233-92-115.ap-south-1.compute.amazonaws.com" -> null

- public_ip = "13.233.92.115" -> null

- secondary_private_ips = [] -> null

- security_groups = [

  - "default",

  ] -> null

- source_dest_check = true -> null

- subnet_id = "subnet-0b62104472025c636" -> null

- tags = {

  - "name" = "Hello-World"

  } -> null

- tags_all = {

  - "name" = "Hello-World"

  } -> null

- tenancy = "default" -> null

- user_data_replace_on_change = false -> null

- vpc_security_group_ids = [

  - "sg-0d901e970546b4e0f",

  ] -> null

- capacity_reservation_specification {

  - capacity_reservation_preference = "open" -> null

  }

```
- cpu_options {
    - core_count       = 1 -> null
    - threads_per_core = 1 -> null
  }

- credit_specification {
    - cpu_credits = "standard" -> null
  }

- enclave_options {
    - enabled = false -> null
  }

- maintenance_options {
    - auto_recovery = "default" -> null
  }

- metadata_options {
    - http_endpoint               = "enabled" -> null
    - http_protocol_ipv6          = "disabled" -> null
    - http_put_response_hop_limit = 2 -> null
    - http_tokens                 = "required" -> null
    - instance_metadata_tags      = "disabled" -> null
  }
```

```
    - private_dns_name_options {
        - enable_resource_name_dns_a_record    = false -> null
        - enable_resource_name_dns_aaaa_record = false -> null
        - hostname_type                        = "ip-name" -> null
      }

    - root_block_device {
        - delete_on_termination = true -> null
        - device_name           = "/dev/xvda" -> null
        - encrypted             = false -> null
        - iops                  = 3000 -> null
        - tags                  = {} -> null
        - throughput            = 125 -> null
        - volume_id             = "vol-03d4da02bdc9ad228" -> null
        - volume_size           = 8 -> null
        - volume_type           = "gp3" -> null
      }
  }

# aws_key_pair.sim_ssh_key will be destroyed
- resource "aws_key_pair" "sim_ssh_key" {
    - arn         = "arn:aws:ec2:ap-south-1:043241213129:key-pair/terra.sim.key" -> null
    - fingerprint = "0a:42:d3:c5:05:87:67:50:5e:8a:d8:94:e5:87:2c:03" -> null
    - id          = "terra.sim.key" -> null
    - key_name    = "terra.sim.key" -> null
```

- key_pair_id = "key-0d500f6346633417c" -> null

- key_type   = "rsa" -> null

- public_key = "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDI1OilC6Ljn/JfviO7VFXoof25l0kXT9NP1+l4Ak8wgqD4GUY3Gnqx
sADYCe+9emQmORMnQaA3V5zj70flkKyXsZsDP9lVLlkgMJDGefcL8VoaN4u+Sxn0qagW8QsQhIFK+pFZr9y
mM4oBIAD8vBcxXYMDzaAvYsj1ZauoZNZRJ/wmY2rnlULD+Zvjw6ips6FH9oRqAgXjMDlWJtTSyRquQunWM
8vuVTd+NbRrY/dFN86HpetYEJAWIeP8EtwfoIUTBTjz9lKoVKLyQNXOGn/x3AFgR167mmXb0li37Dj9BtPJ1z7
31NWOcJqN+tFo98XCLtjU0DWVqEs2fSNWtRPN03pOiYOXRuz6VOzXN7OFsFUkGM4uTG6Sd4YHaCHIaG5
MVIMDG17Fq5jJskqQkZwp6oddVCwAI1nTW6n35mH3/ZNRA3QSnAqcRmoJZE6y3me7Vp4T8Lf03LZIapTI/
HDCP8uYG98NN9hGIGuWwPvlc8ZxSsoazrT0uY6gFrh/kPk=" -> null

- tags   = {} -> null

- tags_all   = {} -> null

}

aws_security_group.sim_terra_sg will be destroyed

- resource "aws_security_group" "sim_terra_sg" {

- arn   = "arn:aws:ec2:ap-south-1:043241213129:security-group/sg-04455219d59c2cdbe" ->
null

- description   = "Managed by Terraform" -> null

- egress   = [] -> null

- id   = "sg-04455219d59c2cdbe" -> null

- ingress   = [

- {

- cidr_blocks   = [

- "0.0.0.0/0",

]

- description   = ""

- from_port   = 443

- ipv6_cidr_blocks = []

```
      - prefix_list_ids  = []
      - protocol        = "tcp"
      - security_groups  = []
      - self            = false
      - to_port         = 443
    },
  - {
      - cidr_blocks     = [
          - "0.0.0.0/0",
        ]
      - description     = ""
      - from_port       = 80
      - ipv6_cidr_blocks = []
      - prefix_list_ids  = []
      - protocol        = "tcp"
      - security_groups  = []
      - self            = false
      - to_port         = 80
    },
  ] -> null
- name                 = "MY-SG01" -> null
- owner_id             = "043241213129" -> null
- revoke_rules_on_delete = false -> null
- tags                 = {} -> null
- tags_all             = {} -> null
```

```
    - vpc_id               = "vpc-0d7f078357cd79872" -> null
  }

Plan: 0 to add, 0 to change, 3 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_key_pair.sim_ssh_key: Destroying... [id=terra.sim.key]
aws_security_group.sim_terra_sg: Destroying... [id=sg-04455219d59c2cdbe]
aws_instance.terraform_inst: Destroying... [id=i-0a29e681c775bdc97]
aws_key_pair.sim_ssh_key: Destruction complete after 0s
aws_security_group.sim_terra_sg: Destruction complete after 0s
aws_instance.terraform_inst: Still destroying... [id=i-0a29e681c775bdc97, 10s elapsed]
aws_instance.terraform_inst: Still destroying... [id=i-0a29e681c775bdc97, 20s elapsed]
aws_instance.terraform_inst: Still destroying... [id=i-0a29e681c775bdc97, 30s elapsed]
aws_instance.terraform_inst: Destruction complete after 30s

Destroy complete! Resources: 3 destroyed.

================================================================================
```