ALTSCHOOL AFRICA

SCHOOL OF ENGINEERING (CLOUD ENGINEERING TRACK)

MONTH 1 ASSESSMENT (SETTING UP TechCorp)

DEPLOYMENT EVIDENCE

1. Terraform plan output

```
(base) ififrank@Frank-Laptop:/mnt/c/Users/olivc/OneDrive/Documents/altschool-barakat-cohort/third-semester/assignments/terraform-assessment$ terraform plan
data.aws_ami.amazon_linux_2: Reading...
data.aws_availability_zones.available: Reading...
data.aws_availability_zones.available: Read complete after 2s [id=us-east-1]
data.aws_ami.amazon_linux_2: Read complete after 3s [id=ami-0156001f0548e90b1]

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_eip.bastion will be created
  + resource "aws_eip" "bastion" {
      + allocation_id        = (known after apply)
      + arn                  = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = "vpc"
      + id                   = (known after apply)
      + instance             = (known after apply)
      + ipam_pool_id         = (known after apply)
      + network_border_group = (known after apply)
      + network_interface    = (known after apply)
      + private_dns          = (known after apply)
      + private_ip           = (known after apply)
      + ptr_record           = (known after apply)
      + public_dns           = (known after apply)
      + public_ip            = (known after apply)
      + public_ipv4_pool     = (known after apply)
      + tags                 = {
          + "Name" = "techcorp-bastion-eip"
        }
      + tags_all             = {
          + "Name" = "techcorp-bastion-eip"
        }
      + vpc                  = (known after apply)
    }

  # aws_eip.nat_1 will be created
  + resource "aws_eip" "nat_1" {
      + allocation_id        = (known after apply)
      + arn                  = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = "vpc"
      + id                   = (known after apply)
      + instance             = (known after apply)
      + ipam_pool_id         = (known after apply)
      + network_border_group = (known after apply)
      + network_interface    = (known after apply)
      + private_dns          = (known after apply)
      + private_ip           = (known after apply)
      + ptr_record           = (known after apply)
```

```
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + cidr_block                = "0.0.0.0/0"
            + nat_gateway_id            = (known after apply)
              # (11 unchanged attributes hidden)
          },
      ]
    + tags             = {
        + "Name" = "techcorp-private-rt-1"
      }
    + tags_all         = {
        + "Name" = "techcorp-private-rt-1"
      }
    + vpc_id           = (known after apply)
  }

# aws_route_table.private_2 will be created
+ resource "aws_route_table" "private_2" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + cidr_block                = "0.0.0.0/0"
            + nat_gateway_id            = (known after apply)
              # (11 unchanged attributes hidden)
          },
      ]
    + tags             = {
        + "Name" = "techcorp-private-rt-2"
      }
    + tags_all         = {
        + "Name" = "techcorp-private-rt-2"
      }
    + vpc_id           = (known after apply)
  }

# aws_route_table.public will be created
+ resource "aws_route_table" "public" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + cidr_block                = "0.0.0.0/0"
            + gateway_id                = (known after apply)
              # (11 unchanged attributes hidden)
          },
      ]
    + tags             = {
```

```
      + "Name" = "techcorp-public-rt"
    }
  + tags_all         = {
      + "Name" = "techcorp-public-rt"
    }
  + vpc_id           = (known after apply)
}

# aws_route_table_association.private_1 will be created
+ resource "aws_route_table_association" "private_1" {
  + id             = (known after apply)
  + route_table_id = (known after apply)
  + subnet_id      = (known after apply)
}

# aws_route_table_association.private_2 will be created
+ resource "aws_route_table_association" "private_2" {
  + id             = (known after apply)
  + route_table_id = (known after apply)
  + subnet_id      = (known after apply)
}

# aws_route_table_association.public_1 will be created
+ resource "aws_route_table_association" "public_1" {
  + id             = (known after apply)
  + route_table_id = (known after apply)
  + subnet_id      = (known after apply)
}

# aws_route_table_association.public_2 will be created
+ resource "aws_route_table_association" "public_2" {
  + id             = (known after apply)
  + route_table_id = (known after apply)
  + subnet_id      = (known after apply)
}

# aws_security_group.bastion will be created
+ resource "aws_security_group" "bastion" {
  + arn                    = (known after apply)
  + description            = "Security group for bastion host"
  + egress                 = [
      + {
          + cidr_blocks      = [
              + "0.0.0.0/0",
            ]
          + description      = "Allow all outbound traffic"
          + from_port        = 0
          + ipv6_cidr_blocks = []
          + prefix_list_ids  = []
          + protocol         = "-1"
          + security_groups  = []
          + self             = false
          + to_port          = 0
        },
```

```
      ]
    + id                     = (known after apply)
    + ingress                = [
        + {
            + cidr_blocks      = [
                + "41.73.1.67/32",
              ]
            + description      = "SSH from your IP"
            + from_port        = 22
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 22
          },
      ]
    + name                   = "techcorp-bastion-sg"
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + revoke_rules_on_delete = false
    + tags                   = {
        + "Name" = "techcorp-bastion-sg"
      }
    + tags_all               = {
        + "Name" = "techcorp-bastion-sg"
      }
    + vpc_id                 = (known after apply)
  }

# aws_security_group.database will be created
+ resource "aws_security_group" "database" {
    + arn                    = (known after apply)
    + description            = "Security group for database server"
    + egress                 = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = "Allow all outbound traffic"
            + from_port        = 0
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "-1"
            + security_groups  = []
            + self             = false
            + to_port          = 0
          },
      ]
    + id                     = (known after apply)
    + ingress                = [
        + {
            + cidr_blocks      = []
            + description      = "PostgreSQL from web servers"
```

```
            + from_port        = 5432
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = (known after apply)
            + self             = false
            + to_port          = 5432
          },
        + {
            + cidr_blocks      = []
            + description      = "SSH from bastion"
            + from_port        = 22
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = (known after apply)
            + self             = false
            + to_port          = 22
          },
      ]
    + name                   = "techcorp-database-sg"
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + revoke_rules_on_delete = false
    + tags                   = {
        + "Name" = "techcorp-database-sg"
      }
    + tags_all               = {
        + "Name" = "techcorp-database-sg"
      }
    + vpc_id                 = (known after apply)
  }

# aws_security_group.web will be created
+ resource "aws_security_group" "web" {
    + arn                    = (known after apply)
    + description            = "Security group for web servers"
    + egress                 = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = "Allow all outbound traffic"
            + from_port        = 0
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "-1"
            + security_groups  = []
            + self             = false
            + to_port          = 0
          },
      ]
    + id                     = (known after apply)
    + ingress                = [
```

```
      + {
          + cidr_blocks      = [
              + "0.0.0.0/0",
            ]
          + description      = "HTTP from anywhere"
          + from_port        = 80
          + ipv6_cidr_blocks = []
          + prefix_list_ids  = []
          + protocol         = "tcp"
          + security_groups  = []
          + self             = false
          + to_port          = 80
        },
      + {
          + cidr_blocks      = [
              + "0.0.0.0/0",
            ]
          + description      = "HTTPS from anywhere"
          + from_port        = 443
          + ipv6_cidr_blocks = []
          + prefix_list_ids  = []
          + protocol         = "tcp"
          + security_groups  = []
          + self             = false
          + to_port          = 443
        },
      + {
          + cidr_blocks      = []
          + description      = "SSH from bastion"
          + from_port        = 22
          + ipv6_cidr_blocks = []
          + prefix_list_ids  = []
          + protocol         = "tcp"
          + security_groups  = (known after apply)
          + self             = false
          + to_port          = 22
        },
    ]
  + name                   = "techcorp-web-sg"
  + name_prefix            = (known after apply)
  + owner_id               = (known after apply)
  + revoke_rules_on_delete = false
  + tags                   = {
      + "Name" = "techcorp-web-sg"
    }
  + tags_all               = {
      + "Name" = "techcorp-web-sg"
    }
  + vpc_id                 = (known after apply)
}

# aws_subnet.private_1 will be created
+ resource "aws_subnet" "private_1" {
    + arn                                            = (known after apply)
```

```
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.3.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "techcorp-private-subnet-1"
        }
      + tags_all                                       = {
          + "Name" = "techcorp-private-subnet-1"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_subnet.private_2 will be created
  + resource "aws_subnet" "private_2" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-1b"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.4.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "techcorp-private-subnet-2"
        }
      + tags_all                                       = {
          + "Name" = "techcorp-private-subnet-2"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_subnet.public_1 will be created
  + resource "aws_subnet" "public_1" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.1.0/24"
```

```
    + enable_dns64                                 = false
    + enable_resource_name_dns_a_record_on_launch  = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                           = (known after apply)
    + ipv6_cidr_block_association_id               = (known after apply)
    + ipv6_native                                  = false
    + map_public_ip_on_launch                      = true
    + owner_id                                     = (known after apply)
    + private_dns_hostname_type_on_launch          = (known after apply)
    + tags                                         = {
        + "Name" = "techcorp-public-subnet-1"
      }
    + tags_all                                     = {
        + "Name" = "techcorp-public-subnet-1"
      }
    + vpc_id                                       = (known after apply)
  }

# aws_subnet.public_2 will be created
+ resource "aws_subnet" "public_2" {
    + arn                                          = (known after apply)
    + assign_ipv6_address_on_creation              = false
    + availability_zone                            = "us-east-1b"
    + availability_zone_id                         = (known after apply)
    + cidr_block                                   = "10.0.2.0/24"
    + enable_dns64                                 = false
    + enable_resource_name_dns_a_record_on_launch  = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                           = (known after apply)
    + ipv6_cidr_block_association_id               = (known after apply)
    + ipv6_native                                  = false
    + map_public_ip_on_launch                      = true
    + owner_id                                     = (known after apply)
    + private_dns_hostname_type_on_launch          = (known after apply)
    + tags                                         = {
        + "Name" = "techcorp-public-subnet-2"
      }
    + tags_all                                     = {
        + "Name" = "techcorp-public-subnet-2"
      }
    + vpc_id                                       = (known after apply)
  }

# aws_vpc.main will be created
+ resource "aws_vpc" "main" {
    + arn                                = (known after apply)
    + cidr_block                         = "10.0.0.0/16"
    + default_network_acl_id             = (known after apply)
    + default_route_table_id             = (known after apply)
    + default_security_group_id          = (known after apply)
    + dhcp_options_id                    = (known after apply)
    + enable_dns_hostnames               = true
    + enable_dns_support                 = true
    + enable_network_address_usage_metrics = (known after apply)
```

```
    + public_dns            = (known after apply)
    + public_ip             = (known after apply)
    + public_ipv4_pool      = (known after apply)
    + tags                  = {
        + "Name" = "techcorp-nat-eip-1"
      }
    + tags_all              = {
        + "Name" = "techcorp-nat-eip-1"
      }
    + vpc                   = (known after apply)
  }

# aws_eip.nat_2 will be created
+ resource "aws_eip" "nat_2" {
    + allocation_id         = (known after apply)
    + arn                   = (known after apply)
    + association_id        = (known after apply)
    + carrier_ip            = (known after apply)
    + customer_owned_ip     = (known after apply)
    + domain                = "vpc"
    + id                    = (known after apply)
    + instance              = (known after apply)
    + ipam_pool_id          = (known after apply)
    + network_border_group  = (known after apply)
    + network_interface     = (known after apply)
    + private_dns           = (known after apply)
    + private_ip            = (known after apply)
    + ptr_record            = (known after apply)
    + public_dns            = (known after apply)
    + public_ip             = (known after apply)
    + public_ipv4_pool      = (known after apply)
    + tags                  = {
        + "Name" = "techcorp-nat-eip-2"
      }
    + tags_all              = {
        + "Name" = "techcorp-nat-eip-2"
      }
    + vpc                   = (known after apply)
  }

# aws_instance.bastion will be created
+ resource "aws_instance" "bastion" {
    + ami                                  = "ami-0156001f0548e90b1"
    + 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)
    + enable_primary_ipv6                  = (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                          = "t3.micro"
+ ipv6_address_count                     = (known after apply)
+ ipv6_addresses                         = (known after apply)
+ key_name                               = "aws-key"
+ 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" = "techcorp-bastion-host"
  }
+ tags_all                               = {
    + "Name" = "techcorp-bastion-host"
  }
+ tenancy                                = (known after apply)
+ user_data                              = (sensitive value)
+ user_data_base64                       = (known after apply)
+ user_data_replace_on_change            = false
+ vpc_security_group_ids                 = (known after apply)

+ capacity_reservation_specification (known after apply)

+ cpu_options (known after apply)

+ ebs_block_device (known after apply)

+ enclave_options (known after apply)

+ ephemeral_block_device (known after apply)

+ instance_market_options (known after apply)

+ maintenance_options (known after apply)

+ metadata_options (known after apply)
```

```
    + network_interface (known after apply)

    + private_dns_name_options (known after apply)

    + root_block_device (known after apply)
  }

# aws_instance.database will be created
+ resource "aws_instance" "database" {
    + ami                                  = "ami-0156001f0548e90b1"
    + 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)
    + enable_primary_ipv6                  = (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                        = "t3.small"
    + ipv6_address_count                   = (known after apply)
    + ipv6_addresses                       = (known after apply)
    + key_name                             = "aws-key"
    + 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" = "techcorp-database-server"
      }
    + tags_all                             = {
        + "Name" = "techcorp-database-server"
      }
    + tenancy                              = (known after apply)
    + user_data                            = "6196830be1c9f2221ca33139412f2bd82838d744"
```

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

    + capacity_reservation_specification (known after apply)

    + cpu_options (known after apply)

    + ebs_block_device (known after apply)

    + enclave_options (known after apply)

    + ephemeral_block_device (known after apply)

    + instance_market_options (known after apply)

    + maintenance_options (known after apply)

    + metadata_options (known after apply)

    + network_interface (known after apply)

    + private_dns_name_options (known after apply)

    + root_block_device (known after apply)
  }

# aws_instance.web_1 will be created
+ resource "aws_instance" "web_1" {
    + ami                                  = "ami-0156001f0548e90b1"
    + 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)
    + enable_primary_ipv6                  = (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                        = "t3.micro"
    + ipv6_address_count                   = (known after apply)
    + ipv6_addresses                       = (known after apply)
    + key_name                             = "aws-key"
    + 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" = "techcorp-web-server-1"
      }
    + tags_all                             = {
        + "Name" = "techcorp-web-server-1"
      }
    + tenancy                              = (known after apply)
    + user_data                            = "0fde8a00dabce3b1a801c3bbe046e25019892101"
    + user_data_base64                     = (known after apply)
    + user_data_replace_on_change          = false
    + vpc_security_group_ids               = (known after apply)

    + capacity_reservation_specification (known after apply)

    + cpu_options (known after apply)

    + ebs_block_device (known after apply)

    + enclave_options (known after apply)

    + ephemeral_block_device (known after apply)

    + instance_market_options (known after apply)

    + maintenance_options (known after apply)

    + metadata_options (known after apply)

    + network_interface (known after apply)

    + private_dns_name_options (known after apply)

    + root_block_device (known after apply)
  }

# aws_instance.web_2 will be created
+ resource "aws_instance" "web_2" {
    + ami                                  = "ami-0156001f0548e90b1"
    + 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)
+ enable_primary_ipv6                   = (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                         = "t3.micro"
+ ipv6_address_count                    = (known after apply)
+ ipv6_addresses                        = (known after apply)
+ key_name                              = "aws-key"
+ 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" = "techcorp-web-server-2"
  }
+ tags_all                              = {
    + "Name" = "techcorp-web-server-2"
  }
+ tenancy                               = (known after apply)
+ user_data                             = "0fde8a00dabce3b1a801c3bbe046e25019892101"
+ user_data_base64                      = (known after apply)
+ user_data_replace_on_change           = false
+ vpc_security_group_ids                = (known after apply)

+ capacity_reservation_specification (known after apply)

+ cpu_options (known after apply)

+ ebs_block_device (known after apply)

+ enclave_options (known after apply)

+ ephemeral_block_device (known after apply)
```

```
    + instance_market_options (known after apply)

    + maintenance_options (known after apply)

    + metadata_options (known after apply)

    + network_interface (known after apply)

    + private_dns_name_options (known after apply)

    + root_block_device (known after apply)
  }

# aws_internet_gateway.main will be created
+ resource "aws_internet_gateway" "main" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + tags     = {
        + "Name" = "techcorp-igw"
      }
    + tags_all = {
        + "Name" = "techcorp-igw"
      }
    + vpc_id   = (known after apply)
  }

# aws_lb.main will be created
+ resource "aws_lb" "main" {
    + arn                                                        = (known after apply)
    + arn_suffix                                                 = (known after apply)
    + client_keep_alive                                          = 3600
    + desync_mitigation_mode                                     = "defensive"
    + dns_name                                                   = (known after apply)
    + drop_invalid_header_fields                                 = false
    + enable_deletion_protection                                 = false
    + enable_http2                                               = true
    + enable_tls_version_and_cipher_suite_headers                = false
    + enable_waf_fail_open                                       = false
    + enable_xff_client_port                                     = false
    + enable_zonal_shift                                         = false
    + enforce_security_group_inbound_rules_on_private_link_traffic = (known after apply)
    + id                                                         = (known after apply)
    + idle_timeout                                               = 60
    + internal                                                   = false
    + ip_address_type                                            = (known after apply)
    + load_balancer_type                                         = "application"
    + name                                                       = "techcorp-alb"
    + name_prefix                                                = (known after apply)
    + preserve_host_header                                       = false
    + security_groups                                            = (known after apply)
    + subnets                                                    = (known after apply)
    + tags                                                       = {
```

```
    + tags                                                       = {
        + "Name" = "techcorp-alb"
      }
    + tags_all                                                   = {
        + "Name" = "techcorp-alb"
      }
    + vpc_id                                                     = (known after apply)
    + xff_header_processing_mode                                 = "append"
    + zone_id                                                    = (known after apply)

    + subnet_mapping (known after apply)
  }

# aws_lb_listener.http will be created
+ resource "aws_lb_listener" "http" {
    + arn                                                                = (known after apply)
    + id                                                                 = (known after apply)
    + load_balancer_arn                                                  = (known after apply)
    + port                                                               = 80
    + protocol                                                           = "HTTP"
    + routing_http_request_x_amzn_mtls_clientcert_header_name            = (known after apply)
    + routing_http_request_x_amzn_mtls_clientcert_issuer_header_name     = (known after apply)
    + routing_http_request_x_amzn_mtls_clientcert_leaf_header_name       = (known after apply)
    + routing_http_request_x_amzn_mtls_clientcert_serial_number_header_name = (known after apply)
    + routing_http_request_x_amzn_mtls_clientcert_subject_header_name    = (known after apply)
    + routing_http_request_x_amzn_mtls_clientcert_validity_header_name   = (known after apply)
    + routing_http_request_x_amzn_tls_cipher_suite_header_name           = (known after apply)
    + routing_http_request_x_amzn_tls_version_header_name                = (known after apply)
    + routing_http_response_access_control_allow_credentials_header_value = (known after apply)
    + routing_http_response_access_control_allow_headers_header_value    = (known after apply)
    + routing_http_response_access_control_allow_methods_header_value    = (known after apply)
    + routing_http_response_access_control_allow_origin_header_value     = (known after apply)
    + routing_http_response_access_control_expose_headers_header_value   = (known after apply)
    + routing_http_response_access_control_max_age_header_value          = (known after apply)
    + routing_http_response_content_security_policy_header_value         = (known after apply)
    + routing_http_response_server_enabled                               = (known after apply)
    + routing_http_response_strict_transport_security_header_value       = (known after apply)
    + routing_http_response_x_content_type_options_header_value          = (known after apply)
    + routing_http_response_x_frame_options_header_value                 = (known after apply)
    + ssl_policy                                                         = (known after apply)
    + tags_all                                                           = (known after apply)
    + tcp_idle_timeout_seconds                                           = (known after apply)

    + default_action {
        + order            = (known after apply)
        + target_group_arn = (known after apply)
        + type             = "forward"
      }

    + mutual_authentication (known after apply)
  }

# aws_lb_target_group.web will be created
+ resource "aws_lb_target_group" "web" {
    + arn                                = (known after apply)
```

```
      + arn_suffix                         = (known after apply)
      + connection_termination             = (known after apply)
      + deregistration_delay               = "300"
      + id                                 = (known after apply)
      + ip_address_type                    = (known after apply)
      + lambda_multi_value_headers_enabled = false
      + load_balancer_arns                 = (known after apply)
      + load_balancing_algorithm_type      = (known after apply)
      + load_balancing_anomaly_mitigation  = (known after apply)
      + load_balancing_cross_zone_enabled  = (known after apply)
      + name                               = "techcorp-web-tg"
      + name_prefix                        = (known after apply)
      + port                               = 80
      + preserve_client_ip                 = (known after apply)
      + protocol                           = "HTTP"
      + protocol_version                   = (known after apply)
      + proxy_protocol_v2                  = false
      + slow_start                         = 0
      + tags                               = {
          + "Name" = "techcorp-web-tg"
        }
      + tags_all                           = {
          + "Name" = "techcorp-web-tg"
        }
      + target_type                        = "instance"
      + vpc_id                             = (known after apply)

      + health_check {
          + enabled             = true
          + healthy_threshold   = 2
          + interval            = 30
          + matcher             = "200"
          + path                = "/"
          + port                = "traffic-port"
          + protocol            = "HTTP"
          + timeout             = 3
          + unhealthy_threshold = 2
        }

      + stickiness (known after apply)

      + target_failover (known after apply)

      + target_group_health (known after apply)

      + target_health_state (known after apply)
    }

  # aws_lb_target_group_attachment.web_1 will be created
  + resource "aws_lb_target_group_attachment" "web_1" {
      + id               = (known after apply)
      + port             = 80
      + target_group_arn = (known after apply)
      + target_id        = (known after apply)
```

```
  }

# aws_lb_target_group_attachment.web_2 will be created
+ resource "aws_lb_target_group_attachment" "web_2" {
    + id               = (known after apply)
    + port             = 80
    + target_group_arn = (known after apply)
    + target_id        = (known after apply)
  }

# aws_nat_gateway.nat_1 will be created
+ resource "aws_nat_gateway" "nat_1" {
    + allocation_id                      = (known after apply)
    + association_id                     = (known after apply)
    + connectivity_type                  = "public"
    + id                                 = (known after apply)
    + network_interface_id               = (known after apply)
    + private_ip                         = (known after apply)
    + public_ip                          = (known after apply)
    + secondary_private_ip_address_count = (known after apply)
    + secondary_private_ip_addresses     = (known after apply)
    + subnet_id                          = (known after apply)
    + tags                               = {
        + "Name" = "techcorp-nat-gateway-1"
      }
    + tags_all                           = {
        + "Name" = "techcorp-nat-gateway-1"
      }
  }

# aws_nat_gateway.nat_2 will be created
+ resource "aws_nat_gateway" "nat_2" {
    + allocation_id                      = (known after apply)
    + association_id                     = (known after apply)
    + connectivity_type                  = "public"
    + id                                 = (known after apply)
    + network_interface_id               = (known after apply)
    + private_ip                         = (known after apply)
    + public_ip                          = (known after apply)
    + secondary_private_ip_address_count = (known after apply)
    + secondary_private_ip_addresses     = (known after apply)
    + subnet_id                          = (known after apply)
    + tags                               = {
        + "Name" = "techcorp-nat-gateway-2"
      }
    + tags_all                           = {
        + "Name" = "techcorp-nat-gateway-2"
      }
  }

# aws_route_table.private_1 will be created
+ resource "aws_route_table" "private_1" {
    + arn              = (known after apply)
    + id               = (known after apply)
```

2. **Terraform apply completion**

3. **AWS Console showing created resources**

4. **Load balancer serving web pages from both instances**

5. **SSH access through bastion host**

6. **SSH access to the Web and DB servers**

7. **Connect to the postgres instance on the DB server**

8. **Web access to the Web servers via the ALB URL**