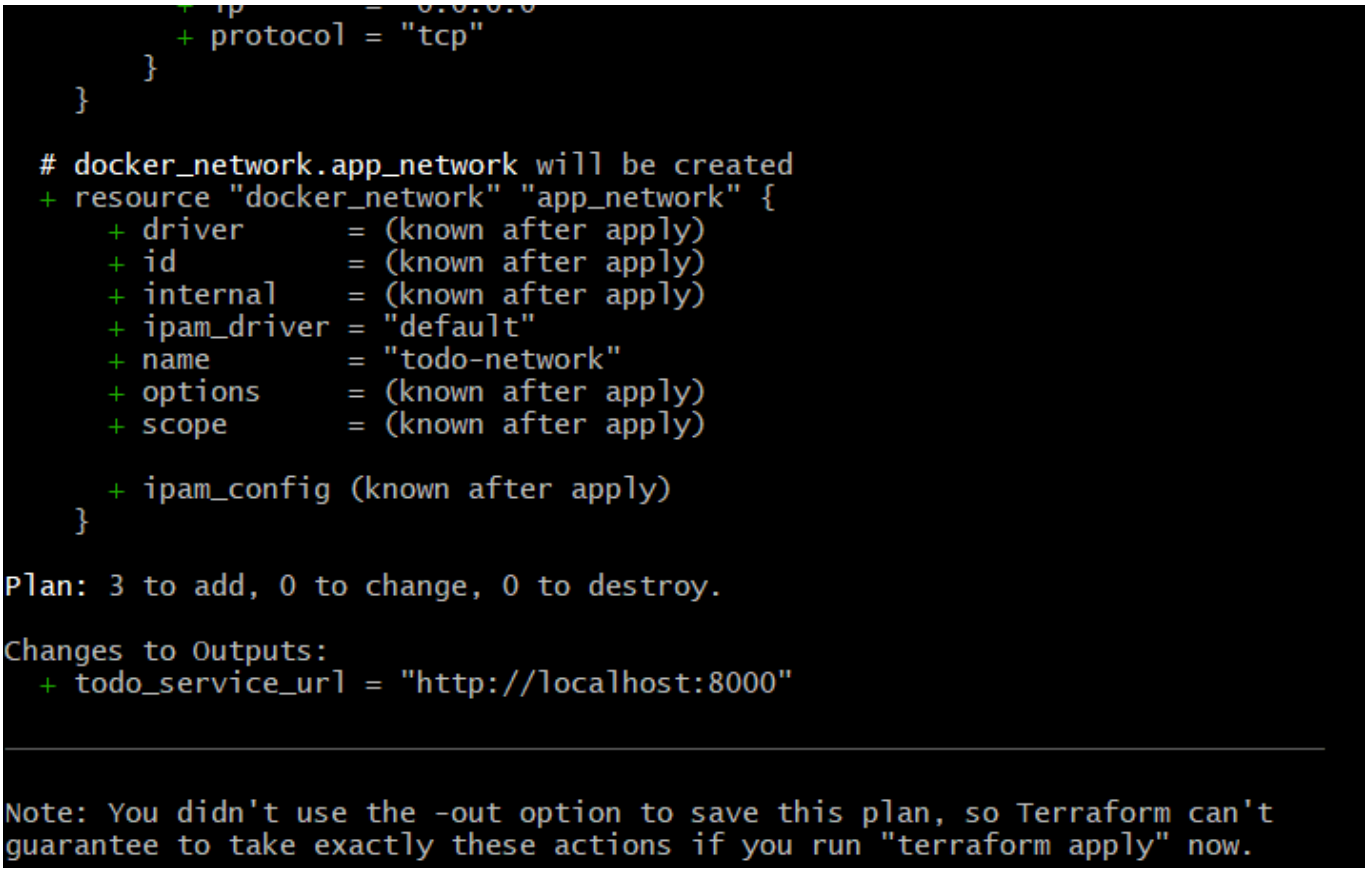
```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "~> 3.0"
    }
  }
}

provider "docker" {}

# Custom network for both services
resource "docker_network" "app_network" {
  name = "todo-network"
}

# Redis container
resource "docker_container" "redis" {
  name  = "redis"
  image = "redis:alpine"

  networks_advanced {
    name = docker_network.app_network.name
  }

  ports {
    internal = 6379
    external = 6379
  }
}

# Todo service container
resource "docker_container" "todo" {
  name  = "todo-service"
  image = "natawut/todo-service:release-2.1"

  networks_advanced {
    name = docker_network.app_network.name
  }

  ports {
    internal = 8000
    external = 8000
  }

  env = [
    "REDIS_HOST=redis"
  ]

  depends_on = [docker_container.redis]
}
```

```
# Output the todo-service URL
output "todo_service_url" {
  value = "http://localhost:8000"
}
```

```
Vivobook@Nacnano-ASUS-Laptop MINGW64 ~/github/my-chula-courses/2110415-soft-def-sys/activity05 (main)
$ "C:\Program Files\Terraform\terraform.exe" init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "~> 3.0"...
- Installing kreuzwerker/docker v3.6.2...
x- Installed kreuzwerker/docker v3.6.2 (self-signed, key ID BD080C4571C6104C)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://developer.hashicorp.com/terraform/cli/plugins/signing
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
```

```
Vivobook@Nacnano-ASUS-Laptop MINGW64 ~/github/my-chula-courses/2110415-soft-def-sys/activity05 (main)
$ "C:\Program Files\Terraform\terraform.exe" plan

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:

  # docker_container.redis will be created
  + resource "docker_container" "redis" {
      + attach                                      = false
      + bridge                                      = (known after apply)
      + command                                     = (known after apply)
      + container_logs                              = (known after apply)
      + container_read_refresh_timeout_milliseconds = 15000
      + entrypoint                                  = (known after apply)
      + env                                         = (known after apply)
      + exit_code                                   = (known after apply)
      + hostname                                    = (known after apply)
      + id                                          = (known after apply)
      + image                                       = "redis:alpine"
      + init                                        = (known after apply)
      + ipc_mode                                    = (known after apply)
      + log_driver                                  = (known after apply)
      + logs                                        = false
      + must_run                                    = true
      + name                                        = "redis"
      + network_data                                = (known after apply)
      + network_mode                                = "bridge"
      + read_only                                   = false
      + remove_volumes                              = true
      + restart                                     = "no"
      + rm                                          = false
      + runtime                                     = (known after apply)
      + security_opts                               = (known after apply)
      + shm_size                                    = (known after apply)
      + start                                       = true
      + stdin_open                                  = false
      + stop_signal                                 = (known after apply)
      + stop_timeout                                = (known after apply)
      + tty                                         = false
      + wait                                        = false
```



```
      + protocol = "tcp"
    }
  }

  # docker_network.app_network will be created
  + resource "docker_network" "app_network" {
      + driver      = (known after apply)
      + id          = (known after apply)
      + internal    = (known after apply)
      + ipam_driver = "default"
      + name        = "todo-network"
      + options     = (known after apply)
      + scope       = (known after apply)

      + ipam_config (known after apply)
    }

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

Changes to Outputs:
  + todo_service_url = "http://localhost:8000"

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.
```

```
Vivobook@Nacnano-ASUS-Laptop MINGW64 ~/github/my-chula-courses/2110415-soft-def-sys/activity05 (main)
$ "C:\Program Files\Terraform\terraform.exe" apply

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:

  # docker_container.redis will be created
  + resource "docker_container" "redis" {
      + attach                                      = false
      + bridge                                      = (known after apply)
      + command                                     = (known after apply)
      + container_logs                              = (known after apply)
      + container_read_refresh_timeout_milliseconds = 15000
      + entrypoint                                  = (known after apply)
      + env                                         = (known after apply)
      + exit_code                                   = (known after apply)
      + hostname                                    = (known after apply)
      + id                                          = (known after apply)
      + image                                       = "redis:alpine"
      + init                                        = (known after apply)
      + ipc_mode                                    = (known after apply)
      + log_driver                                  = (known after apply)
      + logs                                        = false
      + must_run                                    = true
      + name                                        = "redis"
      + network_data                                = (known after apply)
      + network_mode                                = "bridge"
      + read_only                                   = false
      + remove_volumes                              = true
      + restart                                     = "no"
      + rm                                          = false
      + runtime                                     = (known after apply)
      + security_opts                               = (known after apply)
      + shm_size                                    = (known after apply)
      + start                                       = true
      + stdin_open                                  = false
      + stop_signal                                 = (known after apply)
      + stop_timeout                                = (known after apply)
      + tty                                         = false
      + wait                                        = false
```

```
      + scope         = (known after apply)

      + ipam_config (known after apply)
    }

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

Changes to Outputs:
  + todo_service_url = "http://localhost:8000"

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

docker_network.app_network: Creating...
docker_network.app_network: Creation complete after 2s [id=37edd1b02e06aabc475ee
c195873441071cda1d1bdd85e943b12ca54445886f2]
docker_container.redis: Creating...
docker_container.redis: Still creating... [00m10s elapsed]
docker_container.redis: Still creating... [00m20s elapsed]
docker_container.redis: Still creating... [00m30s elapsed]
docker_container.redis: Still creating... [00m40s elapsed]
docker_container.redis: Still creating... [00m50s elapsed]
docker_container.redis: Still creating... [01m00s elapsed]
docker_container.redis: Still creating... [01m10s elapsed]
docker_container.redis: Still creating... [01m20s elapsed]
docker_container.redis: Creation complete after 1m22s [id=a3c9f8d1a43bf0e6b09fe1
9c5bbcc6d16d332a28d8a3a55b12e89bd234cd6cc1]
docker_container.todo: Creating...
docker_container.todo: Still creating... [00m10s elapsed]
docker_container.todo: Still creating... [00m20s elapsed]
docker_container.todo: Still creating... [00m30s elapsed]
docker_container.todo: Still creating... [00m40s elapsed]
docker_container.todo: Still creating... [00m50s elapsed]
docker_container.todo: Still creating... [01m00s elapsed]
docker_container.todo: Still creating... [01m10s elapsed]
docker_container.todo: Still creating... [01m20s elapsed]
docker_container.todo: Still creating... [01m30s elapsed]
docker_container.todo: Still creating... [01m40s elapsed]
docker_container.todo: Still creating... [01m50s elapsed]
docker_container.todo: Still creating... [02m00s elapsed]
docker_container.todo: Still creating... [02m10s elapsed]
docker_container.todo: Still creating... [02m20s elapsed]
docker_container.todo: Still creating... [02m30s elapsed]
docker_container.todo: Still creating... [02m40s elapsed]
docker_container.todo: Still creating... [02m50s elapsed]
docker_container.todo: Still creating... [03m00s elapsed]
docker_container.todo: Still creating... [03m10s elapsed]
```

```
docker_container.todo: Still creating... [05m20s elapsed]
docker_container.todo: Still creating... [05m30s elapsed]
docker_container.todo: Still creating... [05m40s elapsed]
docker_container.todo: Still creating... [05m50s elapsed]
docker_container.todo: Still creating... [06m00s elapsed]
docker_container.todo: Still creating... [06m10s elapsed]
docker_container.todo: Still creating... [06m20s elapsed]
docker_container.todo: Still creating... [06m30s elapsed]
docker_container.todo: Still creating... [06m40s elapsed]
docker_container.todo: Still creating... [06m50s elapsed]
docker_container.todo: Still creating... [07m00s elapsed]
docker_container.todo: Still creating... [07m10s elapsed]
docker_container.todo: Creation complete after 7m15s [id=e146cc9b762b5c929d73536
a9a3c1a413a5f4060b344a646b0f1489d89cc6dbe]

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

Outputs:

todo_service_url = "http://localhost:8000"
```