Installing Terraform

```
serhii@serhii-VirtualBox:~$ curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -
OK
serhii@serhii-VirtualBox:~$ sudo apt-add-repository "deb [arch=$(dpkg --print-architecture)] https://
apt.releases.hashicorp.com $(lsb_release -cs) main"
Hit:1 http://ua.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ua.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ua.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 https://apt.releases.hashicorp.com focal InRelease [16,3 kB]
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:6 https://download.docker.com/linux/ubuntu focal InRelease
Get:7 https://apt.releases.hashicorp.com focal/main amd64 Packages [58,3 kB]
Fetched 74,5 kB in 3s (29,4 kB/s)
Reading package lists... Done
serhii@serhii-VirtualBox:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://ua.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://ua.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://ua.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 https://apt.releases.hashicorp.com focal InRelease
Hit:6 https://download.docker.com/linux/ubuntu focal InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
86 packages can be upgraded. Run 'apt list --upgradable' to see them.
serhii@serhii-VirtualBox:~$ sudo apt-get install terraform
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 86 not upgraded.
Need to get 19,9 MB of archives.
After this operation, 62,9 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com focal/main amd64 terraform amd64 1.2.3 [19,9 MB]
Fetched 19,9 MB in 10s (1 965 kB/s)
Selecting previously unselected package terraform.
(Reading database ... 191795 files and directories currently installed.)
Preparing to unpack .../terraform_1.2.3_amd64.deb ...
Unpacking terraform (1.2.3) ...
Setting up terraform (1.2.3) ...
serhii@serhii-VirtualBox:~$
```

Created main.tf

```
serhii@serhii-VirtualBox:~/project$ ls
docker  main.tf
serhii@serhii-VirtualBox:~/project$ nano main.tf
serhii@serhii-VirtualBox:~/project$ nano main.tf
serhii@serhii-VirtualBox:~/project$ cat main.tf
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "~> 2.13.0"
    }
  }
}

provider "docker" {}

resource "docker_image" "apache2_image" {
  name         = "apache2_image:latest"
  keep_locally = false
}

resource "docker_container" "apache2_image" {
  image = docker_image.apache2_image.latest
  name  = "apache2_image"
  ports {
    internal = 80
    external = 8080
  }
}
```

Initialization Terraform

```
serhii@serhii-VirtualBox:~/project$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "~> 2.13.0"...
- Installing kreuzwerker/docker v2.13.0...
- Installed kreuzwerker/docker v2.13.0 (self-signed, key ID 24E54F214569A8A5)

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://www.terraform.io/docs/cli/plugins/signing.html

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.
serhii@serhii-VirtualBox:~/project$ ls -a
.  ..  docker  main.tf  .terraform  .terraform.lock.hcl
serhii@serhii-VirtualBox:~/project$ terraform validate
Success! The configuration is valid.

serhii@serhii-VirtualBox:~/project$
```

```
serhii@serhii-VirtualBox:~/project$ terraform fmt
serhii@serhii-VirtualBox:~/project$ ls
docker  main.tf
serhii@serhii-VirtualBox:~/project$ cat main.tf
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "~> 2.13.0"
    }
  }
}

provider "docker" {}

resource "docker_image" "apache2_image" {
  name         = "apache2_image:latest"
  keep_locally = false
}

resource "docker_container" "apache2_image" {
  image = docker_image.apache2_image.latest
  name  = "apache2_image"
  ports {
    internal = 80
    external = 8080
  }
}
serhii@serhii-VirtualBox:~/project$
```

Terraform apply

```
serhii@serhii-VirtualBox:~/project$ sudo terraform apply
[sudo] password for serhii:

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.apache2_image will be created
  + resource "docker_container" "apache2_image" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = (known after apply)
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = "json-file"
      + logs             = false
      + must_run         = true
      + name             = "apache2_image"
      + network_data     = (known after apply)
      + read_only        = false
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
      + start            = true
      + stdin_open       = false
```

```
  # docker_image.apache2_image will be created
  + resource "docker_image" "apache2_image" {
      + id           = (known after apply)
      + keep_locally = false
      + latest       = (known after apply)
      + name         = "apache2_image:latest"
      + output       = (known after apply)
      + repo_digest  = (known after apply)
    }

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

docker_image.apache2_image: Creating...
docker_image.apache2_image: Creation complete after 0s [id=sha256:8f2d1c8a0efcf0e8af06e685f6a7d791e5e0a28d29f648935ac2
2c9b9b037e25apache2_image:latest]
docker_container.apache2_image: Creating...
docker_container.apache2_image: Creation complete after 1s [id=5c2515bd34fff6ac871e4623361c73558e648bee860761717a9fd86
06914682f]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
serhii@serhii-VirtualBox:~/project$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                   NAMES
5c2515bd34ff   8f2d1c8a0efc   "/usr/sbin/apache2 -…"   3 minutes ago   Up 3 minutes   0.0.0.0:8080->80/tcp    apache2_i
mage
serhii@serhii-VirtualBox:~/project$
```

Terraform destroy

```
serhii@serhii-VirtualBox:~/project$ sudo terraform destroy
docker_image.apache2_image: Refreshing state... [id=sha256:8f2d1c8a0efcf0e8af06e685f6a7d791e5e0a28d29f648935ac22c9b9b0
37e25apache2_image:latest]
docker_container.apache2_image: Refreshing state... [id=5c2515bd34fff6ac871e4623361c73558e648bee860761717a9fd860691468
2f]

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:

  # docker_container.apache2_image will be destroyed
  - resource "docker_container" "apache2_image" {
      - attach            = false -> null
      - command           = [
          - "/usr/sbin/apache2",
          - "-D",
          - "FOREGROUND",
        ] -> null
      - cpu_shares        = 0 -> null
      - dns               = [] -> null
      - dns_opts          = [] -> null
      - dns_search        = [] -> null
      - entrypoint        = [] -> null
      - env               = [] -> null
      - gateway           = "172.17.0.1" -> null
      - group_add         = [] -> null
      - hostname          = "5c2515bd34ff" -> null
      - id                = "5c2515bd34fff6ac871e4623361c73558e648bee860761717a9fd8606914682f" -> null
      - image             = "sha256:8f2d1c8a0efcf0e8af06e685f6a7d791e5e0a28d29f648935ac22c9b9b037e25" -> null
      - init              = false -> null
      - ip_address        = "172.17.0.2" -> null
      - ip_prefix_length  = 16 -> null
      - ipc_mode          = "private" -> null
      - links             = [] -> null
      - log_driver        = "json-file" -> null
      - log_opts          = {} -> null
      - logs              = false -> null
```

```
  # docker_image.apache2_image will be destroyed
  - resource "docker_image" "apache2_image" {
      - id           = "sha256:8f2d1c8a0efcf0e8af06e685f6a7d791e5e0a28d29f648935ac22c9b9b037e25apache2_image:latest" -
> null
      - keep_locally = false -> null
      - latest       = "sha256:8f2d1c8a0efcf0e8af06e685f6a7d791e5e0a28d29f648935ac22c9b9b037e25" -> null
      - name         = "apache2_image:latest" -> null
    }

Plan: 0 to add, 0 to change, 2 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

docker_container.apache2_image: Destroying... [id=5c2515bd34fff6ac871e4623361c73558e648bee860761717a9fd8606914682f]
docker_container.apache2_image: Destruction complete after 0s
docker_image.apache2_image: Destroying... [id=sha256:8f2d1c8a0efcf0e8af06e685f6a7d791e5e0a28d29f648935ac22c9b9b037e25a
pache2_image:latest]
docker_image.apache2_image: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
```