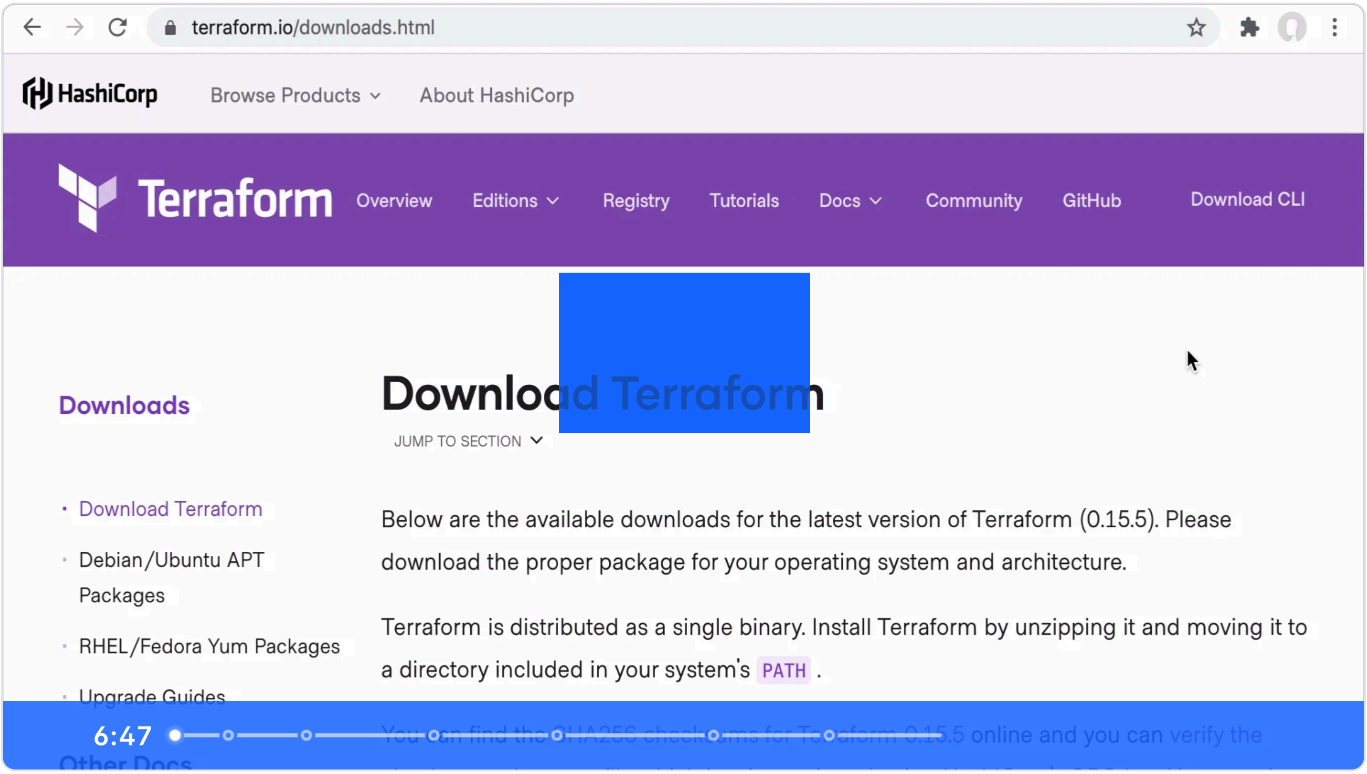
Developer / Terraform / Tutorials / Docker / Install

Install Terraform

9min |

Show Terminal

Reference this often? Create an account to bookmark tutorials.



Downloads

- Download Terraform
- Debian/Ubuntu APT Packages
- RHEL/Fedora Yum Packages
- Upgrade Guides

Other Docs

Download Terraform

JUMP TO SECTION

Below are the available downloads for the latest version of Terraform (0.15.5). Please download the proper package for your operating system and architecture.

Terraform is distributed as a single binary. Install Terraform by unzipping it and moving it to a directory included in your system's PATH .

You can find the SHA256 checksums for Terraform 0.15.5 online and you can verify the

To use Terraform you will need to install it. HashiCorp distributes Terraform as a binary package. You can also install Terraform using popular package managers.

Install Terraform

Manual installation | Homebrew on macOS | Chocolatey on Windows | Linux

Retrieve the `terraform` binary by downloading a pre-compiled binary or compiling it from source.

Pre-compiled binary   Compile from source

To install Terraform, find the appropriate package for your system and download it as a zip archive.

After downloading Terraform, unzip the package. Terraform runs as a single binary named `terraform`. Any other files in the package can be safely removed and Terraform will still function.

Finally, make sure that the `terraform` binary is available on your `PATH`. This process will differ depending on your operating system.

Mac or Linux   Windows

Print a colon-separated list of locations in your `PATH`.

```
$ echo $PATH
```

Move the Terraform binary to one of the listed locations. This command assumes that the binary is currently in your downloads folder and that your `PATH` includes `/usr/local/bin`, but you can customize it if your locations are different.

```
$ mv ~/Downloads/terraform /usr/local/bin/
```

For more detail about adding binaries to your path, see this Stack Overflow article.

Verify the installation

Verify that the installation worked by opening a new terminal session and listing Terraform's available subcommands.

```
$ terraform -help
Usage: terraform [-version] [-help] <command> [args]

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.
##...
```

Add any subcommand to `terraform -help` to learn more about what it does and available options.

```
$ terraform -help plan
```

Troubleshoot

If you get an error that `terraform` could not be found, your `PATH` environment variable was not set up properly. Please go back and ensure that your `PATH` variable contains the directory where Terraform was installed.

Enable tab completion

If you use either Bash or Zsh, you can enable tab completion for Terraform commands. To enable autocomplete, first ensure that a config file exists for your chosen shell.

Bash Zsh

```
$ touch ~/.bashrc
```

Then install the autocomplete package.

```
$ terraform -install-autocomplete
```

Once the autocomplete support is installed, you will need to restart your shell.

Quick start tutorial

Now that you've installed Terraform, you can provision an NGINX server in less than a minute using Docker on Mac, Windows, or Linux. You can also follow the rest of this tutorial in your web browser.

Click on the tab(s) below relevant to your operating system.

Docker Desktop for Mac

Download Docker Desktop for Mac.

After you install Terraform and Docker on your local machine, start Docker Desktop.

```
$ open -a Docker
```

Create a directory named `learn-terraform-docker-container`.

```
$ mkdir learn-terraform-docker-container
```

This working directory houses the configuration files that you write to describe the infrastructure you want Terraform to create and manage. When you initialize and apply the configuration here, Terraform uses this directory to store required plugins, modules (pre-written configurations), and information about the real infrastructure it created.

Navigate into the working directory.

```
$ cd learn-terraform-docker-container
```

In the working directory, create a file called `main.tf` and paste the following Terraform configuration into it.

Mac or Linux   Windows

```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "~> 3.0.1"
    }
  }
}

provider "docker" {}

resource "docker_image" "nginx" {
  name         = "nginx"
  keep_locally = false
}

resource "docker_container" "nginx" {
  image = docker_image.nginx.image_id
  name  = "tutorial"

  ports {
    internal = 80
    external = 8000
  }
}
```

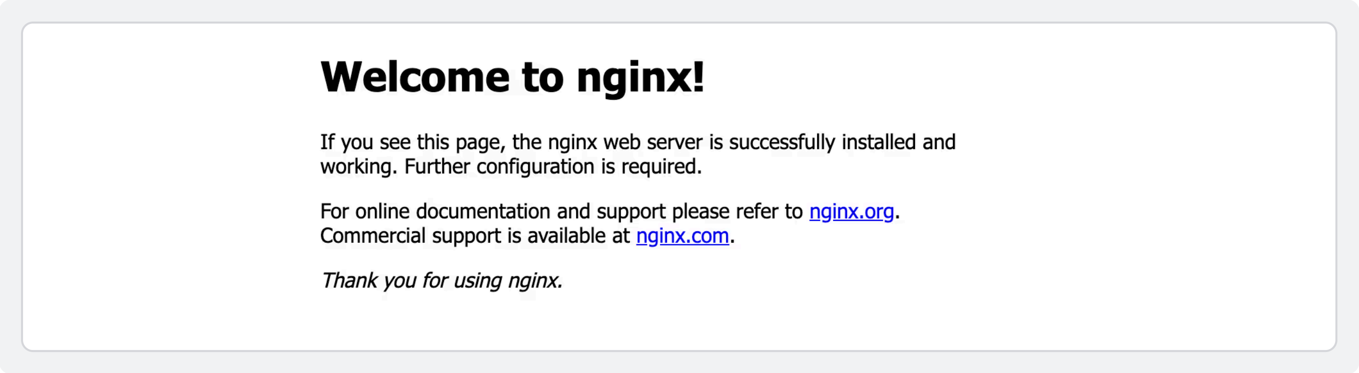
Initialize the project, which downloads a plugin called a provider that lets Terraform interact with Docker.

```
$ terraform init
```

Provision the NGINX server container with `apply`. When Terraform asks you to confirm type `yes` and press `ENTER`.

```
$ terraform apply
```

Verify the existence of the NGINX container by visiting localhost:8000 in your web browser or running `docker ps` to see the container.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

```
$ docker ps
CONTAINER ID        IMAGE                         COMMAND                   CREATED
425d5ee58619        e791337790a6                  "nginx -g 'daemon of…"    20 seconds ago
```

To stop the container, run `terraform destroy`.

```
$ terraform destroy
```

You've now provisioned and destroyed an NGINX webserver with Terraform.

Next Steps

Next, you will create real infrastructure in the cloud of your choice.

- Amazon Web Services (AWS)
- Azure
- Google Cloud Platform (GCP)
- Oracle Cloud Platform (OCI)

Was this tutorial helpful? Yes No

Previous

Infrastructure as Code

Next

Build

This tutorial also appears in:

7 tutorials

Get Started - OCI

Build, change, and destroy a virtual cloud network and subnet on Oracle Cloud Infrastructure (OCI) using Terraform. Step-...

7 tutorials

Get Started - Google Cloud

Build, change, and destroy Google Cloud Platform (GCP) infrastructure using Terraform. Step-by-step, command-line...

8 tutorials

Get Started - Azure

Build, change, and destroy Azure infrastructure using Terraform. Step-by-step, command-line tutorials will walk you throug...

8 tutorials

Get Started - AWS

Build, change, and destroy AWS infrastructure using Terraform. Step-by-step, command-line tutorials will walk you throug...

HashiCorp

Theme System

Certifications System Status Cookie Manager Terms of Use Security Privacy

Trademark Policy Trade Controls Accessibility Give Feedback