

comment installer-terraform sur-ubuntu-server-22-04

Terraform is an open-source infrastructure automation tool that allows you to deploy and manage hundreds of servers via a command-line interface. With Terraform, you can build, change and manage your infrastructure by defining configurations in a human-readable file. It supports many cloud providers, such as AWS, Azure, GCP, and Alibaba cloud. Terraform allows you to automate and provision servers, databases, firewalls, load balancers, and more.

This tutorial will explain how to install Terraform on Ubuntu 22.04.

Prerequisites

- A server running Ubuntu 22.04.
- A root password is configured on the server.

Install Terraform from the Ubuntu Repository

By default, the Terraform package is unavailable in the Ubuntu 22.04 default repository. So, you will need to add the Terraform official repository to your system.

First, add all the required dependencies using the following command:

```
apt install -y gnupg software-properties-common curl -y
```

After installing all the dependencies, add the Terraform GPG key and repository using the following command:

```
curl -fsSL https://apt.releases.hashicorp.com/gpg | apt-key add -  
apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) main"
```

Once the repository is added, you can install the Terraform package with the following command:

```
apt install terraform -y
```

After the successful installation, verify the Terraform version using the following command:

```
terraform --version
```

You should see the following output:

```
Terraform v1.3.6  
on linux_amd64
```

Install Terraform from Source

You can also install Terraform by downloading the latest version from their official website.

You can download it with the following command:

```
wget https://releases.hashicorp.com/terraform/1.3.6/terraform_1.3.6_linux_amd64.zip
```

Once the download is completed, unzip the downloaded file with the following command:

```
unzip terraform_1.3.6_linux_amd64.zip
```

Next, move the Terraform binary to the system location:

```
mv terraform /usr/bin/
```

You can now verify the Terraform version using the following command:

```
terraform --version
```

You will get the following output:

```
Terraform v1.3.6  
on linux_amd64
```

Install Terraform "auto-complete" Feature

The "auto-complete" allows you to see all Terraform sub-commands after pressing the TAB key. You can install this feature with the following command.

```
terraform -install-autocomplete
```

To activate the "auto-complete" feature, logout and log in again to your terminal shell or run the following command:

```
source ~/.bashrc
```

Next, run the Terraform command and press the TAB key twice to verify the "auto-complete" feature.

```
terraform
```

You should see all sub-command in the following output:

```
apply      env        get        init       output    push      state     untaint   workspace
console    fmt       graph     login     plan     refresh  taint     validate
destroy    force-unlock import    logout    providers show      test      version
```

How to Use Terraform

To use the Terraform, first create a directory for your project:

```
mkdir aws
```

Next, navigate to your project directory and create a Terraform configuration file:

```
cd aws
nano main.tf
```

Add the following code:

```
# Provider
provider "aws" {
  access_key = ""
  secret_key = ""
  region = "us-west-1"
}
```

Save and close the file then initialize a Terraform with the following command:

```
terraform init
```

This will automatically download the provider configuration to the .terraform directory.

```
Initializing the backend...
```

```
Initializing provider plugins...
```

```
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.46.0...
- Installed hashicorp/aws v4.46.0 (signed by HashiCorp)
```

```
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.
```

Next, you will need to define resource definitions, set AWS variables then generate and display an execution plan.

You can generate it with the following command:

```
terraform plan
```

Finally, build and deploy your Infrastructure using the following command:

```
terraform apply
```

If you want to destroy the Terraform-managed infrastructure, run the following command:

```
terraform destroy
```

Conclusion

In this post, we showed you how to install Terraform on Ubuntu 22.04. We also explained how to use Terraform to provision AWS instances. You can now create your own Terraform configuration file as per your cloud providers. For more information, visit the Terraform [documentation](#) page.
