



**Laboratorio**  
INSTALACIONES DE RECURSOS  
Terraform



## CONTROL DE VERSIONES

<b>Elaborado por:</b> Jonatan Stiven Gutierrez	<b>No. de Versión:</b> 1.0.0
<b>Revisado por:</b>	<b>Fecha de revisión:</b>
<b>Aprobado por:</b>	<b>Fecha de Aprobación:</b>

### Historia de Modificaciones

No. de Versión	Fecha de Versión	Autor	Revisado por	Aprobado por	Descripción
1.0.0	21/02/2024	Jonatan Stiven Gutierrez			Documento Original

### Lista de distribución

Para	Acción*	Empresa	Firma/Medio de Entrega

\* Tipos de acción: Aprobar, Revisar, Informar, Archivar, Complementar, Asistir a junta, Otras (por favor especificar)

**Este documento fue elaborado por SETI. Prohibida su reproducción total o parcial sin previa autorización del autor.**



## Contenido

<b>INTRODUCCION .....</b>	<b>4</b>
<b>PRERREQUISITOS .....</b>	<b>4</b>
<b>TERRAFORM: .....</b>	<b>5</b>
<b>VISUAL STUDIO CODE: .....</b>	<b>6</b>
<b>INSTALACION TERRAFORM WINDOWS: .....</b>	<b>7</b>
<b>INSTALACION TERRAFORM LINUX: .....</b>	<b>9</b>
<b>CONCLUSION: .....</b>	<b>13</b>



## INTRODUCCION

El siguiente documento proporciona una introducción detallada a Terraform, incluyendo los pasos necesarios para su instalación en sistemas operativos Windows y Linux. Además, abarca la instalación de Visual Studio Code, una herramienta popular para desarrolladores, que facilita la creación y gestión de infraestructuras como código.

## PRERREQUISITOS

- En caso de tener Windows como sistema operativo, pero no contar con permisos de usuario, tener instalada y configurada una máquina virtual.
- Tener conocimientos sobre comandos Windows o Linux.



## TERRAFORM:

Terraform es una herramienta de infraestructura como código (IaC) desarrollada por HashiCorp. Permite a los usuarios definir y provisionar infraestructura de manera declarativa utilizando un lenguaje de configuración simple y legible. Con Terraform, puedes describir tu infraestructura como un conjunto de recursos interrelacionados, como máquinas virtuales, redes, almacenamiento, y más, en un archivo de configuración conocida como "código de infraestructura".

Algunas características clave de Terraform incluyen:

1. **Declaratividad:** Describe la infraestructura deseada en un archivo de configuración en lugar de escribir secuencias de comandos imperativas.
2. **Control de versión:** El código de infraestructura de Terraform se puede almacenar en sistemas de control de versiones como Git, lo que permite el seguimiento de cambios, la colaboración y la reversión a versiones anteriores si es necesario.
3. **Provisionamiento y gestión de estado:** Terraform realiza un seguimiento del estado de tu infraestructura y puede planificar y aplicar cambios de manera segura y automatizada.
4. **Multi-cloud:** Terraform es compatible con varios proveedores de servicios en la nube, como AWS, Azure, Google Cloud Platform, y otros, lo que te permite gestionar la infraestructura en múltiples nubes de forma consistente.
5. **Modularidad:** Puedes organizar tu código de infraestructura en módulos reutilizables, lo que facilita la creación y gestión de configuraciones complejas.

En resumen, Terraform es una herramienta poderosa que ayuda a automatizar la creación, modificación y gestión de la infraestructura de manera eficiente y consistente, lo que facilita la implementación y operación de aplicaciones en la nube y entornos de infraestructura modernos.



## VISUAL STUDIO CODE:

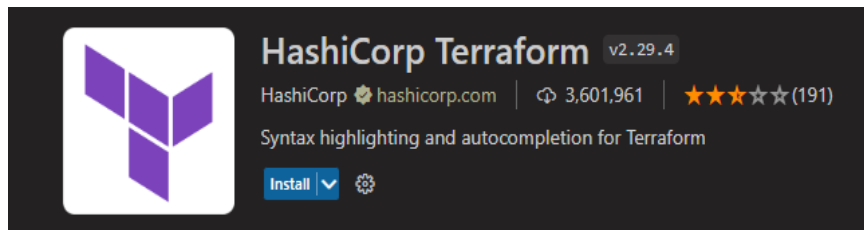
Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es gratuito, de código abierto y está disponible para Windows, macOS y Linux. Es muy popular entre los desarrolladores debido a su amplia gama de características y su extensibilidad.

Página de descarga: [Visual Studio Code - Code](#)

### Extensiones en Visual Studio Code

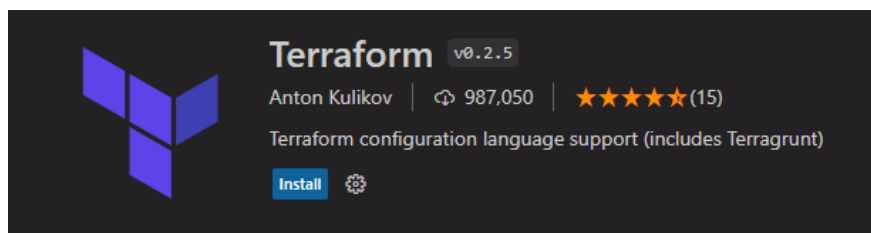
Visual Studio Code (VS Code) ofrece una amplia gama de extensiones que amplían y mejoran su funcionalidad básica. Estas extensiones pueden incluir desde mejoras en la edición de código hasta soporte para lenguajes específicos, integración con herramientas de desarrollo, temas personalizados y mucho más.

- Terraform – HashiCorp Terraform



La extensión HashiCorp Terraform para Visual Studio Code (VS Code) con Terraform Language Server agrega funciones de edición para archivos Terraform, como resaltado de sintaxis, IntelliSense, navegación de código, formato de código, explorador de módulos y mucho más.

- Terraform – Anton Kulikov



Esta es una extensión de Visual Studio Code. Agrega soporte de sintaxis para el lenguaje de configuración Terraform y Terragrunt.

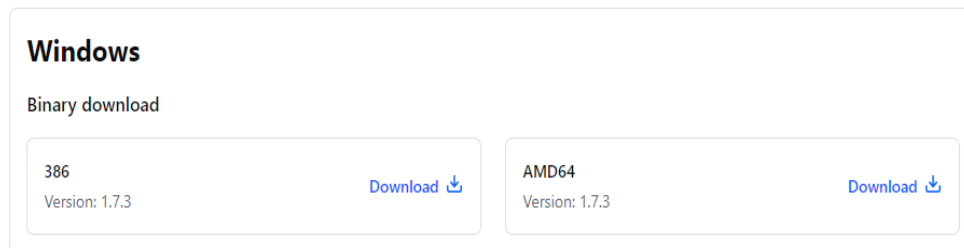


## INSTALACION TERRAFORM WINDOWS:



**Nota:** Portátiles Seti no deja por permisos de administrador.

Link de descarga: <https://developer.hashicorp.com/terraform/install#windows>

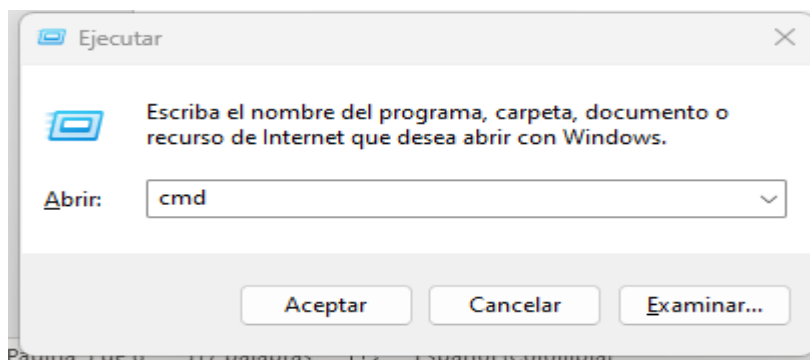
- Ingresamos al link de descarga, vamos a la opción de Windows.
- Seleccionamos de acuerdo a su versión de bits.



- Una vez descargada, creamos una carpeta y en ella movemos el archivo de la descarga.
- Descomprimos el archivo de la descarga.
- Al descomprimir nos genera un archivo terraform.exe, Abrimos el terraform.exe.

Nombre	Fecha de modificación	Tipo	Tamaño
 terraform.exe	19/02/2024 12:12 a. m.	Aplicación	84.370 KB
 terraform_1.7.3_windows_amd64.zip	19/02/2024 12:11 a. m.	Carpeta comprimi...	25.653 KB

- Presionamos la tecla Windows + R, En abrir escribimos cmd, y le damos aceptar.



**Este documento fue elaborado por SETI. Prohibida su reproducción total o parcial sin previa autorización del autor.**



- Se abre la consola de Windows.

```
C:\Windows\system32\cmd.e X + v
Microsoft Windows [Versión 10.0.22621.3155]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Stiven>
```

- Escribimos el comando `cd` hasta llegar a la carpeta previamente creada donde se descomprimos el archivo descargado. Estando en la carpeta donde está el archivo descomprimido, escribimos el comando `terraform.exe --version`. Este comando lo que hace es ejecutar el archivo `.exe` y darnos la version.

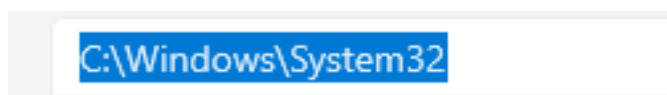
```
D:\Desktop\terraform>terraform.exe --version
Terraform v1.7.3
on windows_amd64
```

- Ahora, terraform solo está instalado en la carpeta creada previamente. Probamos en una carpeta diferente y no reconoce el comando.

```
D:\Desktop\terraform>cd ..

D:\Desktop>terraform.exe --version
"terraform.exe" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
```

- El siguiente paso es instalar terraform de manera global en el sistema. Copiamos el archivo ejecutable `terraform.exe`, lo copiamos, y lo pegamos en la carpeta donde tenga la carpeta de Windows / System32.



**Nota:** Se necesitan permisos de administrador.





- Una vez copiado ahí, ya tendríamos Terraform de manera global en el sistema.

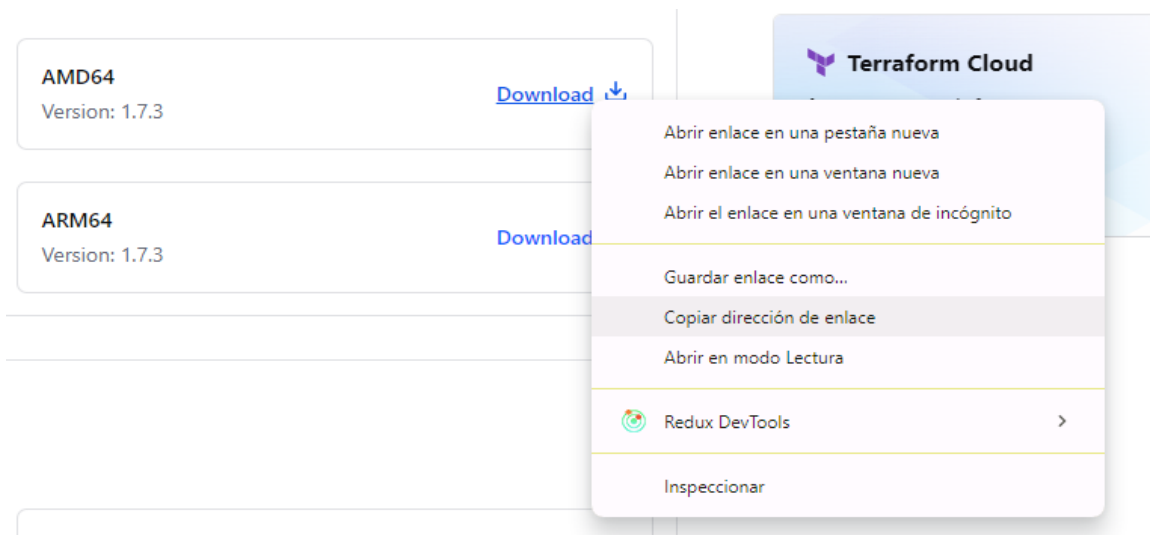
```
D:\Desktop>terraform.exe --version
Terraform v1.7.3
on windows_amd64
```

## INSTALACION TERRAFORM LINUX:

**Nota:** En este ejemplo se hará la instalación en una máquina virtual en Ubuntu 22.04.

Link de descarga: [Install | Terraform | HashiCorp Developer](#)

- Ingresamos al link de descarga, vamos a la opción de Linux.
- Seleccionamos de acuerdo a su versión de bits.
- Versión de bits. 386: 32 bits - AMD64: 64 bits, en este ejemplo será con AMD64.
- Damos click derecho en la opción "Download", y seleccionamos copiar dirección de enlace.





- Entramos en la máquina virtual, abrimos la consola.



- Ahora ejecutamos el siguiente comando.
  - Sudo su, para ingresar el usuario ROOT.
  - Pwd, para saber en qué dirección del directorio estamos.
  - Ls, para saber los archivos del lugar en que estamos.
  - Wget  
[https://releases.hashicorp.com/terraform/1.7.3/terraform\\_1.7.3\\_linux\\_amd64.zip](https://releases.hashicorp.com/terraform/1.7.3/terraform_1.7.3_linux_amd64.zip)  
Este comando es para descargar el archivo Terraform directamente desde consola.





- Una vez descargado.

```
ubuntu [Corriendo] - Oracle VM VirtualBox
19 de feb 14:33
-: sudo su — Konsole

jonatan@ubuntu:/$ sudo su
root@ubuntu:/# pwd
/
root@ubuntu:/# ls
bin  cdrom  docker-compose.yaml  home  lib32  libx32  media  opt  prueba.txt  run  snap  swapfile  var
boot  dev  etc  lib  lib64  lost+found  mnt  proc  root /sbin  srv  sys  usr
root@ubuntu:/# wget https://releases.hashicorp.com/terraform/1.7.3/terraform_1.7.3_linux_amd64.zip
--2024-02-19 14:30:28-- https://releases.hashicorp.com/terraform/1.7.3/terraform_1.7.3_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 18.155.248.45, 18.155.248.90, 18.155.248.119, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|18.155.248.45|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25940403 (25M) [application/zip]
Saving to: 'terraform_1.7.3_linux_amd64.zip'

terraform_1.7.3_linux_amd64.zip      100%[=====] 24,74M  11,0MB/s  in 2,2s

2024-02-19 14:30:30 (11,0 MB/s) - 'terraform_1.7.3_linux_amd64.zip' saved [25940403/25940403]

root@ubuntu:/#
```

- Hacemos un comando LS, para mirar los archivos del lugar que estamos ubicados en el repositorio y validamos la creación de un archivo .zip nuevo.

Ahora hacemos un unzip al archivo .zip.

Comando ls para validar el archivo que se genera, y se generó un archivo terraform.

```
ubuntu [Corriendo] - Oracle VM VirtualBox
19 de feb 14:34
-: sudo su — Konsole

jonatan@ubuntu:/$ sudo su
root@ubuntu:/# pwd
/
root@ubuntu:/# ls
bin  cdrom  docker-compose.yaml  home  lib32  libx32  media  opt  prueba.txt  run  snap  swapfile  terraform_1.7.3_linux_amd64.zip  var
boot  dev  etc  lib  lib64  lost+found  mnt  proc  root /sbin  srv  sys  usr
root@ubuntu:/# wget https://releases.hashicorp.com/terraform/1.7.3/terraform_1.7.3_linux_amd64.zip
--2024-02-19 14:30:28-- https://releases.hashicorp.com/terraform/1.7.3/terraform_1.7.3_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 18.155.248.45, 18.155.248.90, 18.155.248.119, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|18.155.248.45|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25940403 (25M) [application/zip]
Saving to: 'terraform_1.7.3_linux_amd64.zip'

terraform_1.7.3_linux_amd64.zip      100%[=====] 24,74M  11,0MB/s  in 2,2s

2024-02-19 14:30:30 (11,0 MB/s) - 'terraform_1.7.3_linux_amd64.zip' saved [25940403/25940403]

root@ubuntu:/# ls
bin  cdrom  docker-compose.yaml  home  lib32  libx32  media  opt  prueba.txt  run  snap  swapfile  terraform_1.7.3_linux_amd64.zip  var
boot  dev  etc  lib  lib64  lost+found  mnt  proc  root /sbin  srv  sys  usr
root@ubuntu:/# unzip terraform_1.7.3_linux_amd64.zip
Archive: terraform_1.7.3_linux_amd64.zip
  inflating: terraform

root@ubuntu:/# ls
bin  cdrom  docker-compose.yaml  home  lib32  libx32  media  opt  prueba.txt  run  snap  swapfile  terraform  terraform_1.7.3_linux_amd64.zip  var
boot  dev  etc  lib  lib64  lost+found  mnt  proc  root /sbin  srv  sys  usr
```



- Ahora lo ejecutamos con el comando `./terraform -version`

```
ubuntu [Corriendo] - Oracle VM VirtualBox
Actividades  Konsole  19 de feb 14:42
-- sudo su -- Konsole

root@ubuntu:/# ls
bin  cdrom  docker-compose.yaml  home  lib32  libx32  media  opt  prueba.txt  run  snap  swapfile  terraform  var
boot  dev  etc  lib  lib64  lost+found  mnt  proc  root  sbin  srv  sys  terraform_1.7.3_linux_amd64.zip  usr

root@ubuntu:/# ./terraform --version
Terraform v1.7.3
on linux_amd64
root@ubuntu:/#
```

- Terraform quedo instalado en el lugar donde hicimos `./terraform -version`, ahora queremos usar terraform en otro lugar del sistema, pero no funciona.

```
ubuntu [Corriendo] - Oracle VM VirtualBox
Actividades  Konsole  19 de feb 14:42
-- sudo su -- Konsole

root@ubuntu:/# ls
bin  cdrom  docker-compose.yaml  home  lib32  libx32  media  opt  prueba.txt  run  snap  swapfile  terraform  var
boot  dev  etc  lib  lib64  lost+found  mnt  proc  root  sbin  srv  sys  terraform_1.7.3_linux_amd64.zip  usr

root@ubuntu:/# ./terraform --version
Terraform v1.7.3
on linux_amd64
root@ubuntu:/# cd /etc/
root@ubuntu:/etc# ./terraform --version
bash: ./terraform: No such file or directory
root@ubuntu:/etc#
```

- Para solucionar esto, debemos hacer una copia en `/usr/bin/`.  
`cp terraform /usr/bin/`

```
root@ubuntu:/etc# cd ..
root@ubuntu:/# ls
bin  cdrom  docker-compose.yaml  home  lib32  libx32  media  opt  prueba.txt  run  snap  swapfile  terraform  var
boot  dev  etc  lib  lib64  lost+found  mnt  proc  root  sbin  srv  sys  terraform_1.7.3_linux_amd64.zip  usr

root@ubuntu:/# cp terraform /usr/bin/
root@ubuntu:/#
```

- Cambiamos de ruta y probamos de nuevo el comando `terraform -version`.

```
ubuntu [Corriendo] - Oracle VM VirtualBox
Actividades  Konsole

root@ubuntu:/# cd home
root@ubuntu:/home# ls
jonatan
root@ubuntu:/home# terraform --version
Terraform v1.7.3
on linux_amd64
root@ubuntu:/home#
```



## CONCLUSION:

Usar Terraform desde Windows y desde una máquina virtual Ubuntu ofrecen experiencias ligeramente diferentes, pero ambas tienen sus ventajas y desventajas. Aquí hay una conclusión general sobre cada plataforma:

### Terraform desde Windows:

- Ventajas:
  - **Compatibilidad con herramientas de desarrollo de Windows:** Si estás acostumbrado a trabajar en un entorno de desarrollo de Windows y utilizas herramientas específicas de Windows, como Visual Studio Code u otras, trabajar con Terraform desde Windows puede integrarse de manera más fluida con tu flujo de trabajo existente.
  - **Facilidad de configuración inicial:** Terraform se puede instalar fácilmente en Windows utilizando el instalador MSI o la versión portátil, lo que facilita la configuración inicial y la gestión de dependencias.
- Desventajas:
  - **Compatibilidad limitada con ciertas características:** Algunas características avanzadas de Terraform pueden tener una compatibilidad limitada en entornos de Windows en comparación con Linux, lo que podría resultar en restricciones en ciertos escenarios de uso.
  - **Problemas de interoperabilidad:** Si tu infraestructura objetivo está basada en sistemas operativos Linux, podrías enfrentar problemas de interoperabilidad al trabajar con Terraform desde Windows, especialmente en lo que respecta al formato de archivos y la ejecución de scripts.

### Terraform desde una máquina virtual Ubuntu:

- Ventajas:
  - **Compatibilidad total con características avanzadas:** Al trabajar desde una máquina virtual Ubuntu, tendrás compatibilidad total con todas las características avanzadas de Terraform, así como con las

Este documento fue elaborado por SETI. Prohibida su reproducción total o parcial sin previa autorización del autor.



herramientas y servicios relacionados que están más alineados con el ecosistema de Linux.

- **Interoperabilidad mejorada:** Si tu infraestructura objetivo se basa en sistemas operativos Linux, trabajar desde una máquina virtual Ubuntu proporciona una mejor interoperabilidad y coherencia en el flujo de trabajo, lo que puede simplificar la gestión y el mantenimiento a largo plazo.
- Desventajas:
  - **Configuración inicial más compleja:** Configurar una máquina virtual Ubuntu y configurar Terraform puede requerir un poco más de esfuerzo inicial en comparación con instalar Terraform directamente en Windows.
  - **Requiere recursos adicionales de hardware:** Ejecutar una máquina virtual Ubuntu en tu sistema Windows requerirá recursos adicionales de hardware, como memoria RAM y espacio en disco, lo que puede afectar el rendimiento general de tu sistema.

En resumen, la elección entre usar Terraform desde Windows o desde una máquina virtual Ubuntu dependerá de tus preferencias personales, tus habilidades técnicas y los requisitos específicos de tu proyecto. Ambas opciones son viables y pueden proporcionar una experiencia de desarrollo efectiva, así que elige la que mejor se adapte a tus necesidades y circunstancias.